快捷键终极使用指南 | 效率思维

Minja　01月02日

一个工具的入门用户和进阶玩家，区别在哪些地方？相信不少人的回答里会有「会用快捷键」这一项。

我们印象中，在键盘上运指如飞的人，技术基本要比拿鼠标爬格子的人要厉害；即使在键鼠齐上的游戏玩家眼中，全键盘实现避让、瞄准、开枪的选手也是枪林高手。**会用快捷键，往往代表形成了一套肌肉记忆，意味着专业、熟练。**

但是换上自己真刀真枪开始学快捷键了，我们就会遇到很多问题：

1. 快捷键凭什么设计成今天这样子?
2. 怎样才能记住快捷键?
3. 怎样自定义快捷键才不容易发生冲突?
4. 总是不小心按到 `⌘Command-Q` 退出应用，怎样避免?
5. ……

这一篇文章，将从快捷键的历史开始谈起，带你了解为何今日的快捷键设计成这样；通用的东西不一定恰好适合你，所以，知其所以然之后，我们还要通过系统设置、Keyboard Maestro、Karabiner-Elements 等工具和方法，将快捷键变得更加客制化，让它真正符合你的习惯。

文章较长，设计的系统和工具也较多，大家可以先从目录中

挑自己最感兴趣的读。

注：由于我（Minja）只有 Mac 电脑，所以本文主要以 macOS 为例，但原理是通用的。Windows 部分由 @沨沄极客 编写，希望能让尽可能多的读者受益。

快捷键简史

如果平时已经在留意快捷键相关的技巧，你可能听说过一个概念：**Hyper 键**。

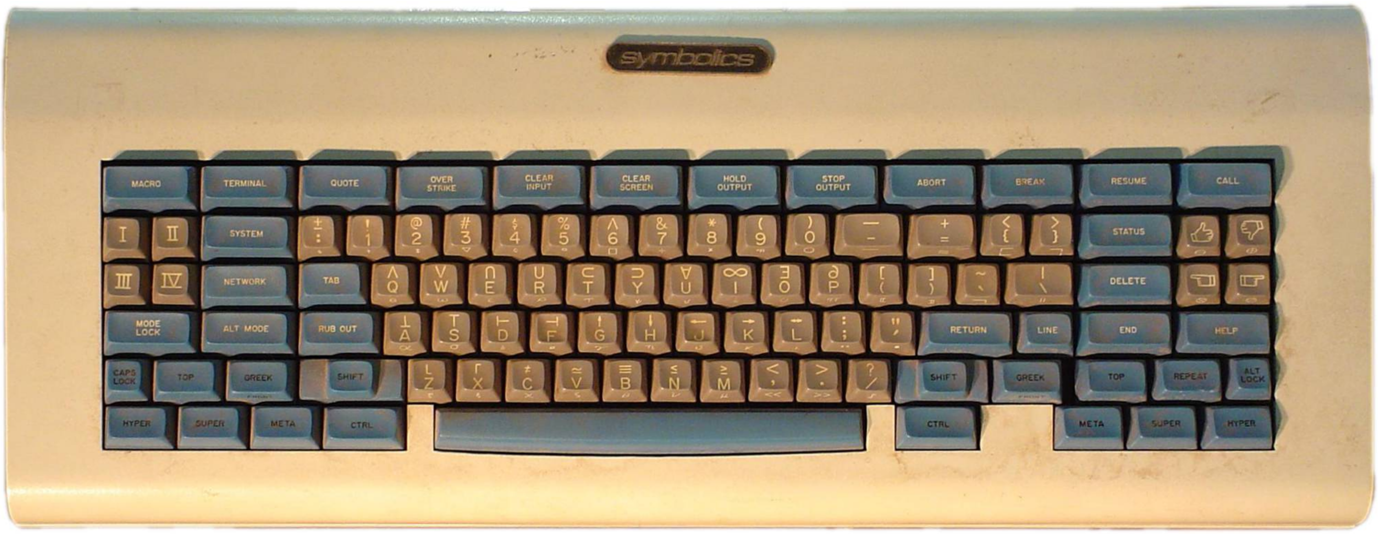
Hyper 键不是一颗真正存在于键盘上的按键，而是 `⇧Shift`、`^Control`、`⌥Option`、`⌘Command` 几颗键组成的「超级按键」。**它不容易和现有的快捷键冲突**[1]

，所以喜欢自定义快捷键的玩家会把自己常用的功能绑定在 Hyper 键上，便于随时触发。

iTerm2
Shell
Edit
View
Profiles
Toolbelt
Window
Help
43°
11:41

通过 Hyper 键召唤出从天而降的终端，图/潜水的科技猫

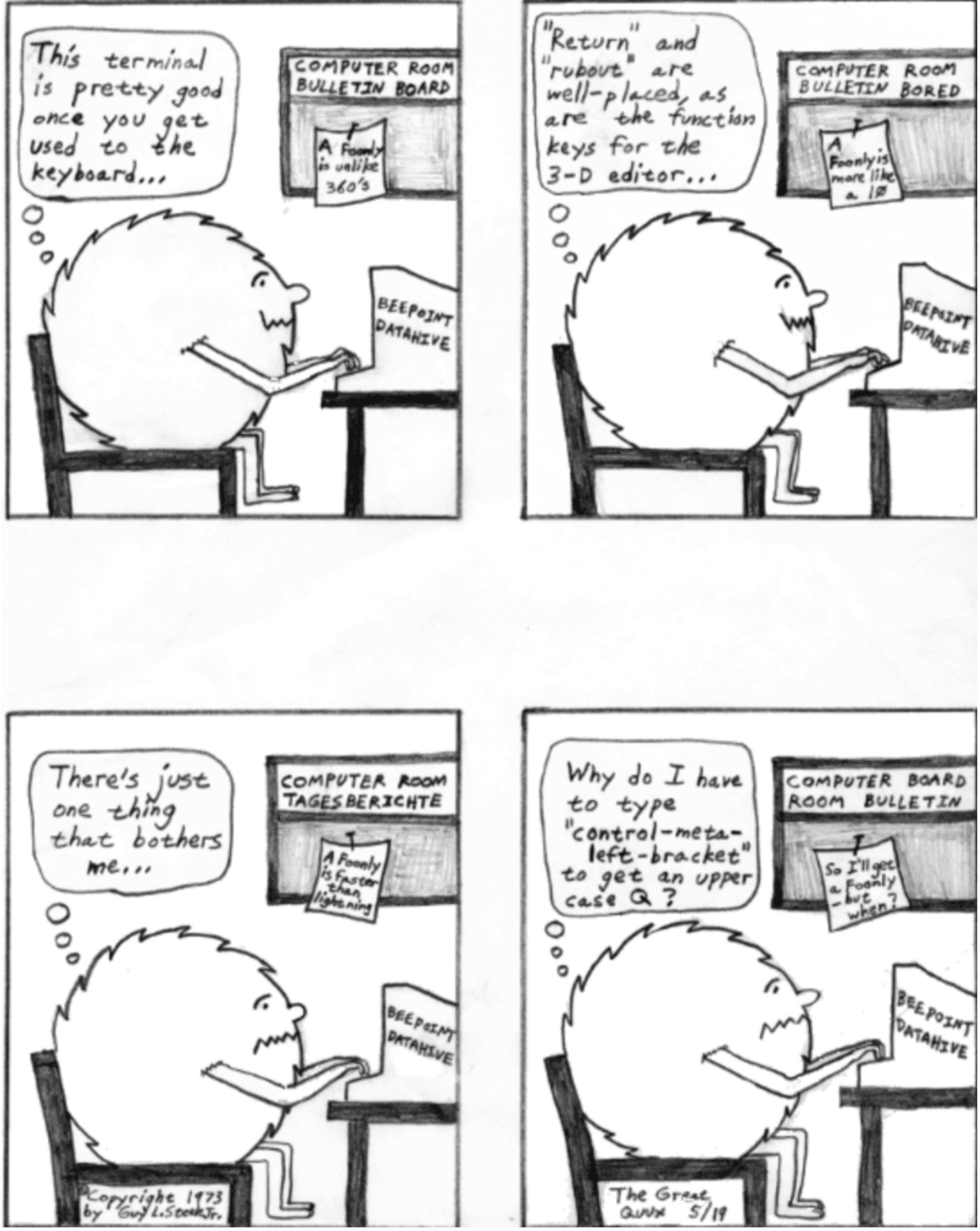
然而在早期，这颗 Hyper 键真的存在。


symbolics
MACRO
TERMINAL
QUOTE
OVER STRIKE
CLEAR INPUT
CLEAR SCREEN
HOLD OUTPUT
STOP OUTPUT
ABORT
BREAK
RESUME
CALL
SYSTEM
STATUS
NETWORK
TAB
DELETE
MODE LOCK
ALT MODE
RUB OUT
RETURN
LINE
END
HELP
CAPS LOCK
TOP
GREEK
SHIFT
SHIFT
GREEK
TOP
REPEAT
ALT LOCK
HYPER
SUPER
META
CTRL
CTRL
META
SUPER
HYPER

Space-cadet

上图是 Space Cadet 键盘，上世纪的麻省理工土特产，它直接影响了「神级」文本编辑器——Emacs 的键位设计。可以看到，Space Cadet 的左下角就是一颗 Hyper 键，紧邻着还有一颗 Super 键。继续数过去，还有 Meta 键以及我们熟悉的 Ctrl 键。再算上上面一排，整整有 7 颗修饰键。这键盘要是落到游戏玩家手里，一定会把他们高兴坏了：可以设置好几颗放大招用的键呢！

早期键盘上那一大堆修饰键，用途和今天差不多，**也是和字母/数字键组合起来用，改变字母/数字键按下后的效果**。不过这些零零碎碎的按键在设计时考虑得显然不够周到，有时会出现常用功能需要好几颗按键一起按才能实现的尴尬状况。

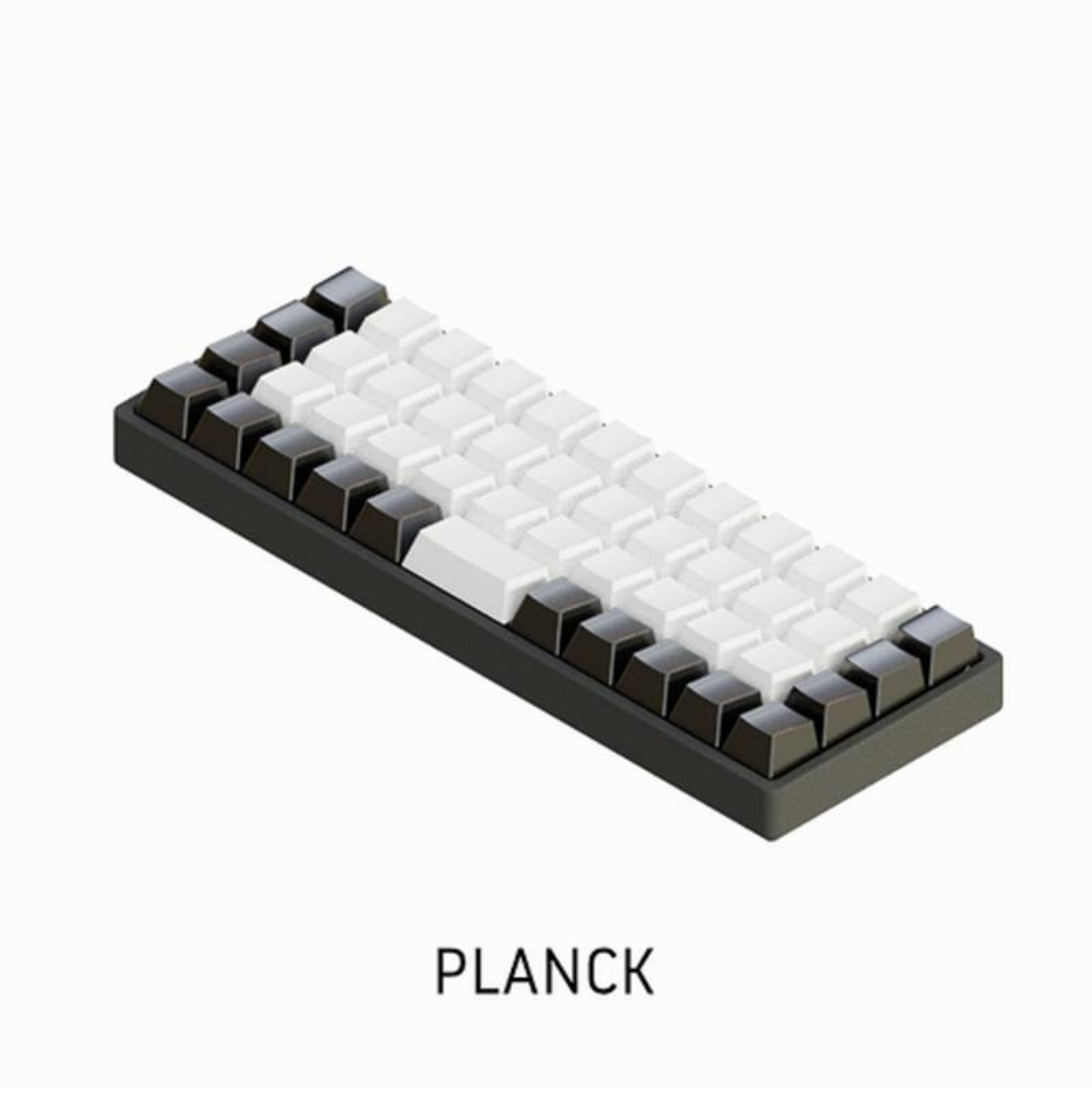

This terminal is pretty good once you get used to the keyboard...
COMPUTER ROOM BULLETIN BOARD
A Foonly is unlike 360's
BEEPOINT DATAHIVE
"Return" and "rubout" are well-placed, as are the function keys for the 3-D editor...
COMPUTER ROOM BULLETIN BORED
A Foonly is more like a 10
BEEPOINT DATAHIVE
There's just one thing that bothers me...
COMPUTER ROOM TAGESBERICHTE
A Foonly is faster than lightning
BEEPOINT DATAHIVE
Copyright 1973 by Guy L. Steele Jr.
Why do I have to type "control-meta-left-bracket" to get an upper case Q?
COMPUTER BOARD ROOM BULLETIN
So I'll get a Foonly – but when?
BEEPOINT DATAHIVE
The Great Quux 5/19

对于 space cadet 键盘的吐槽 图/新黑客词典

后来不知是嫌那么多按键造起来贵，还是真的出于键位优化考虑，现代的键盘一般只剩下 4 颗修饰键（Mac 电脑上是 `⇧Shift`、`^Control`、`⌥Option` 和 `⌘Command`）；而被砍掉的 Hyper、Super 等键，其功能则落到了剩下的按键身上，需要两颗以上的修饰键一起按，来达到原先一颗按键的效果。

如果只用系统的那点功能，这倒也没事，还可以带来体积更

小的产品。但组合复杂起来之后，就很难记住预设的快捷键；另一方面，每次想自定义快捷键，就发现键位被已有的快捷键占用了，挺让人沮丧。小众市场上甚至出现了可以完全自定义按键的键盘，但显然不是每个人都会买回来自己折腾。



没有刻字的 Planck 键盘

总结起来，在今天使用快捷键，我们遇到的就是下面两类问

题：

1. 已有的快捷键，怎么记?
2. 想自设快捷键，怎么操作?

下面，我们将分两部分来分别回答两个问题。最后还有一节我个人的操作经验分享，暂时不知道归入什么流派，但用起来效果不错，就独立介绍。

理解快捷键的关键：Modifier Key

主流的操作系统和应用，其快捷键设计都有一套规律，最常见的就是**「修饰键（Modifier Key） + 字母键/数字键/Fn 功能键」**。

组合一个快捷键时，字母、数字和功能键的选择比较简单，一般就是选首字母，不够用了就往后取，再不够就按大小取数字，至于常用功能也会直接绑上功能键（比如 `F12` 表示调高音量）。基本看一眼按键上的符号，然后用上几次，就能够记住。当然，此时我们要切换到英文环境下进行思考，毕竟 `⌘Command-Q` 中的 `Q` 被公认为代表了 `Quit` 的首字母，而不是 `去你的` 的第一个拼音字母。

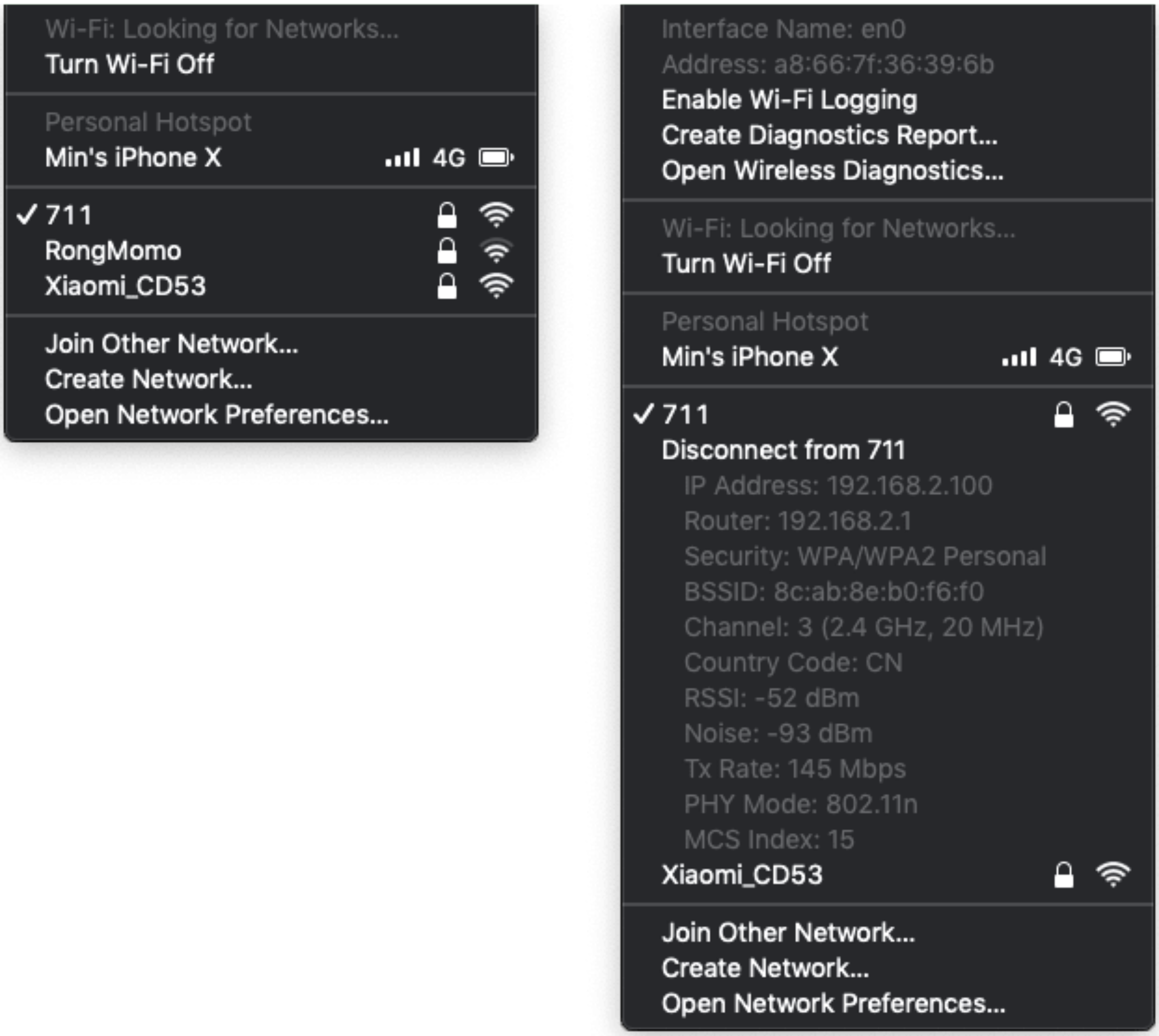
真正难记的，是修饰键。`⌘Command` 还好说，加上一个 `⌥Option` 作为补充也能接受，那 `⇧Shift`、`⌃Control` 又是怎么回事？为什么 Windows 下的 `⌃Control` 和 macOS 中的 `⌃Control` 功能截然不同？**慢慢地，我们就要触及修饰键的意义了。**

本节将一览两大主流操作系统中的快捷键设计，来解决本文的第一大问题：**怎么记快捷键**。

macOS 中的系统快捷键

Mac 键盘上有 4 颗主要的修饰键，官方 给出的功能清单是按应用分的，洋洋洒洒一长串。不过，我们可以摸出一个大致规律：

- `⌘Command`：**命令键**，可以直接和字母/数字键组合成各种常见快捷键。
- `⌥Option`：**选择键**，当 `⌘Command` 系列不够用时作为补充，常常是原有功能的变体，比如 `⌘Command-C` 变成 `⌥Option-⌘Command-C` 后，对应的操作就从拷贝文件变成了拷贝文件路径。另外，按住 `⌥Option` 的同时再点开一些菜单，也可以看到更多功能的变体。



Wi-Fi: Looking for Networks...
Turn Wi-Fi Off
Personal Hotspot
Min's iPhone X 4G
✓ 711
RongMomo
Xiaomi_CD53
Join Other Network...
Create Network...
Open Network Preferences...

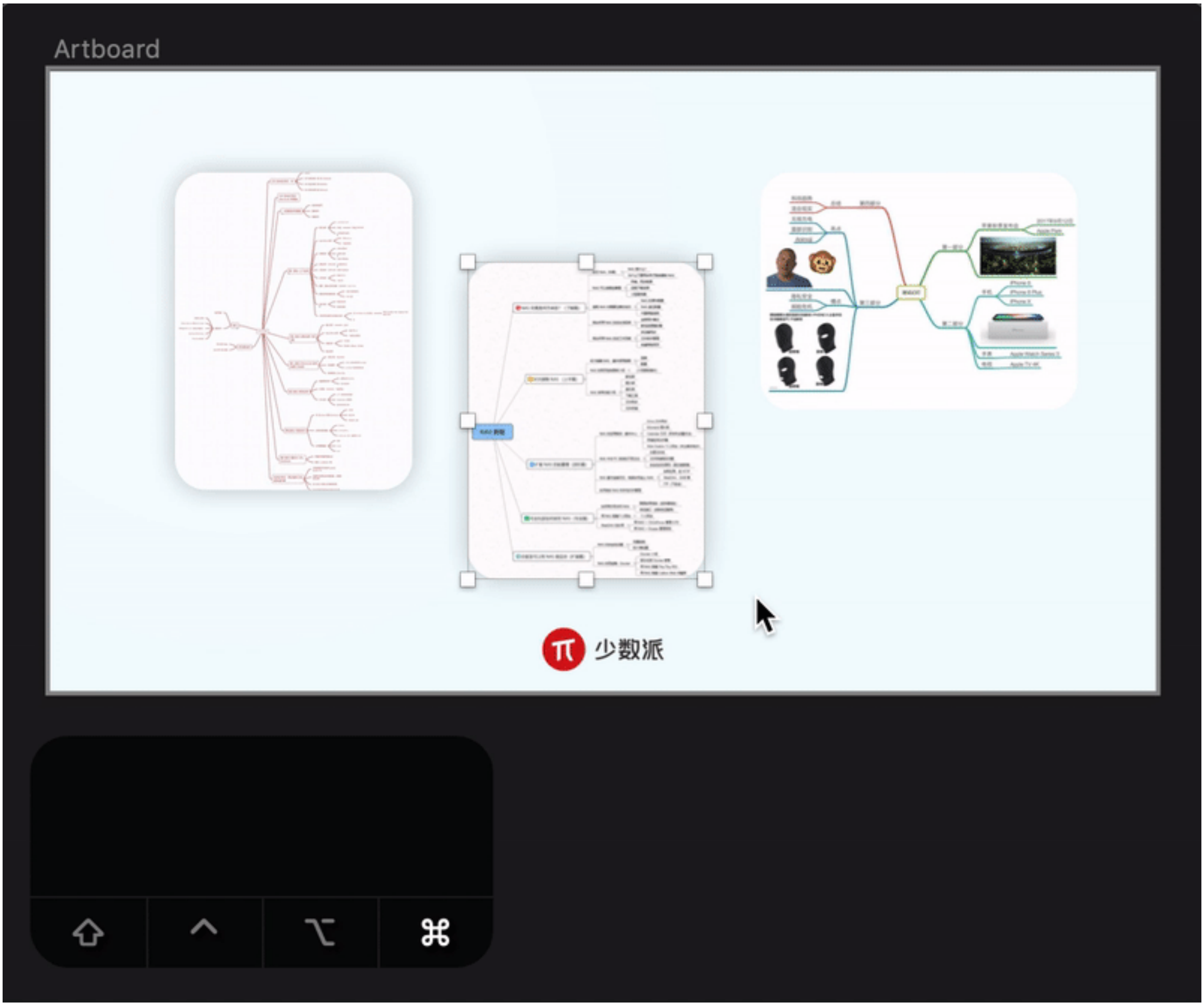
Interface Name: en0
Address: a8:66:7f:36:39:6b
Enable Wi-Fi Logging
Create Diagnostics Report...
Open Wireless Diagnostics...
Wi-Fi: Looking for Networks...
Turn Wi-Fi Off
Personal Hotspot
Min's iPhone X 4G
✓ 711
Disconnect from 711
IP Address: 192.168.2.100
Router: 192.168.2.1
Security: WPA/WPA2 Personal
BSSID: 8c:ab:8e:b0:f6:f0
Channel: 3 (2.4 GHz, 20 MHz)
Country Code: CN
RSSI: -52 dBm
Noise: -93 dBm
Tx Rate: 145 Mbps
PHY Mode: 802.11n
MCS Index: 15
Xiaomi_CD53
Join Other Network...
Create Network...
Open Network Preferences...

按住 ⌥Option 键后可以看到更丰富的菜单选项

- `⇧Shift`：**换档键**，对上面两个按键再进行补充，有时也表示反选、反向操作，比如浏览网页时 `⇧Shift-空格` 就是反向滚动（往上滚）。
- `^Control`：**控制键**，继承自 Emacs 编辑器，一般用在文本操作中。偶尔也作为 `⌥Option`、`⇧Shift` 之后第三梯队的补充。

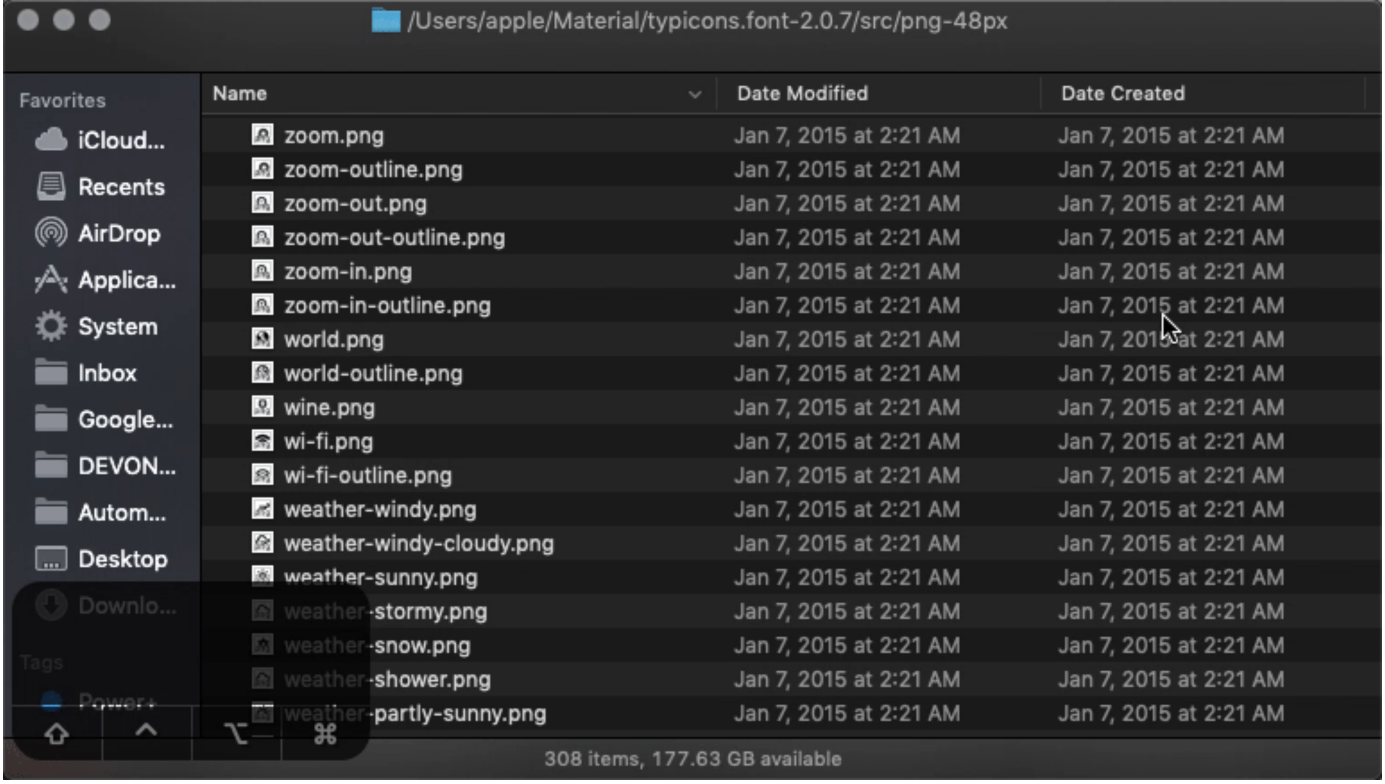
这样捋下来，除去 `⇧Shift-⌘Command-3` 这个谜语一样的截屏快捷键，其他快捷键就比较好记，包括第三方工具的快捷键也可以推算出来。比如在不少设计工具中，`⌘Command-C` 当然

是拷贝某个部件，而 `⌥Option-⌘Command-C` 则是拷贝样式（之后 `⌥Option-⌘Command-V` 就是粘贴样式了）。这样只需绘制一个模板，就可以重复利用样式。



Sketch 中通过快捷键拷贝样式

`⇧Shift` 键还有其他的惊喜。有些读者工作时习惯外接显示器、配上鼠标来工作，此时若遇上一个很宽的网页、一张很大的表格或者一横排文件数据，就没法像触控板那样随意拖动画面。其实，可以按住 `⇧Shift` 再滚动鼠标滚轮，就能横向移动了。



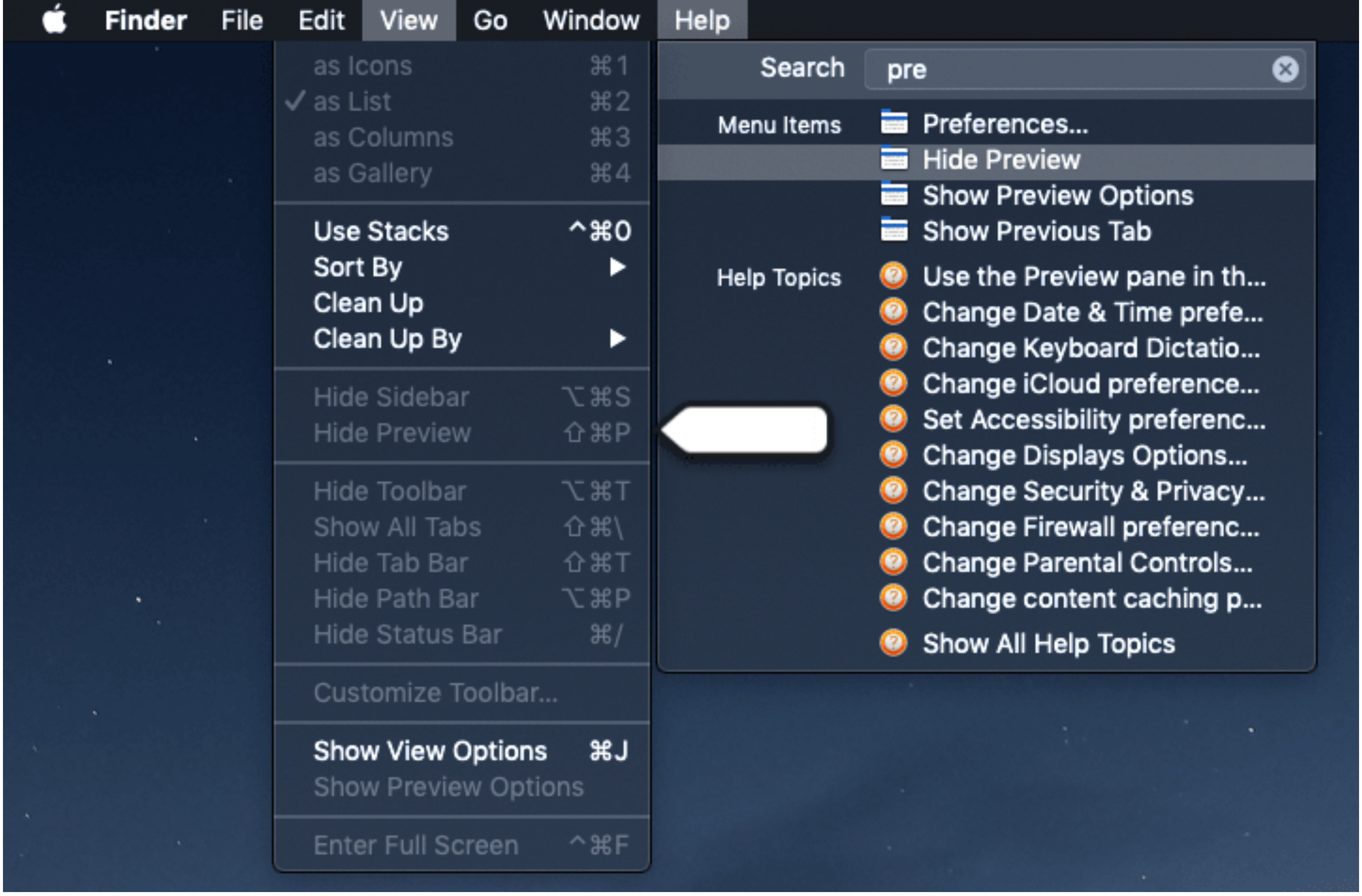
/Users/apple/Material/typicons.font-2.0.7/src/png-48px

Name	Date Modified	Date Created
zoom.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
zoom-outline.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
zoom-out.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
zoom-out-outline.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
zoom-in.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
zoom-in-outline.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
world.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
world-outline.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
wine.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
wi-fi.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
wi-fi-outline.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-windy.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-windy-cloudy.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-sunny.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-stormy.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-snow.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-shower.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM
weather-partly-sunny.png	Jan 7, 2015 at 2:21 AM	Jan 7, 2015 at 2:21 AM

308 items, 177.63 GB available

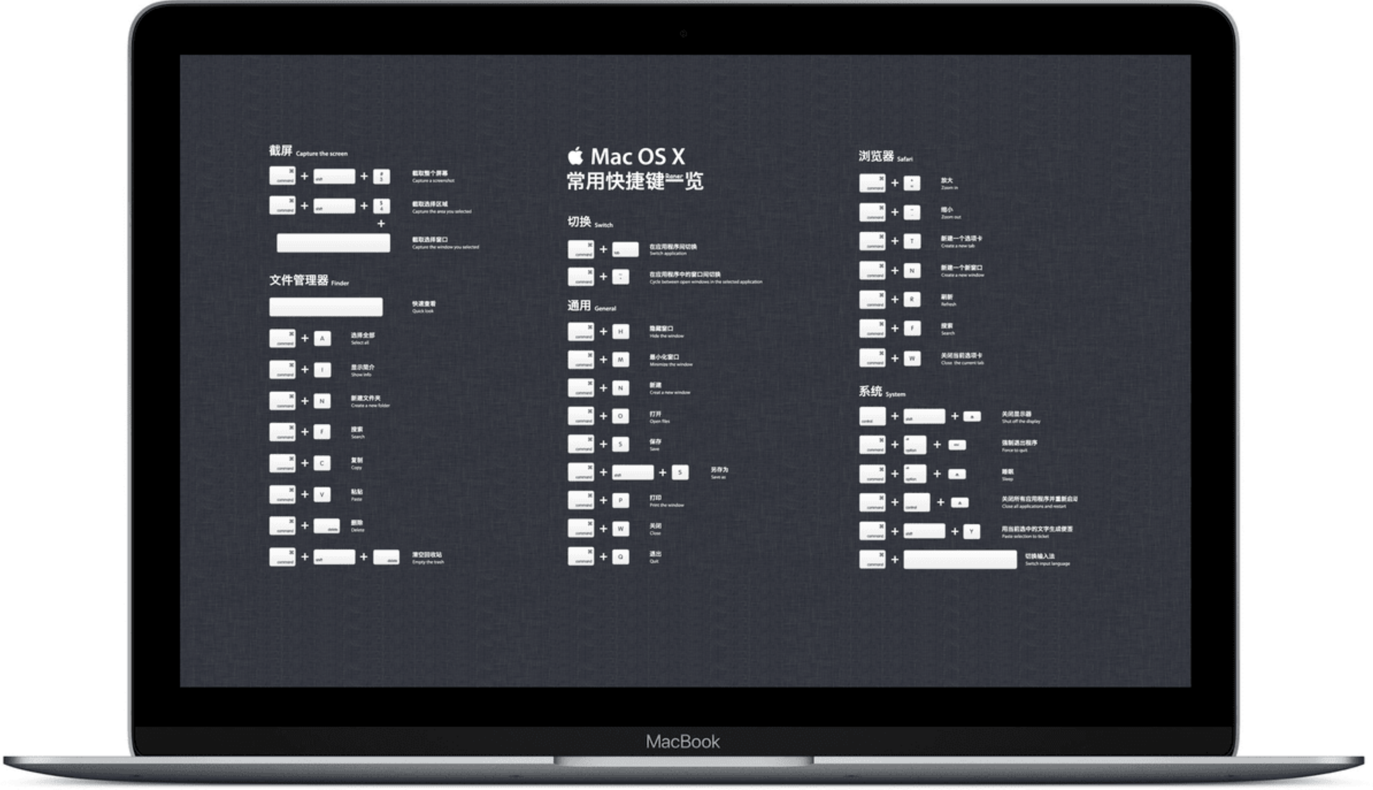
按住 ⇧Shift 键就可以横向滚动

具体到日常习惯上，使用中不妨经常留意菜单栏中的选择，各个功能的快捷键在其中都有标出；不想各个菜单挨个翻过去的话，还可以打开「Help - Seaerch」，找找你想要的功能对应了哪些快捷键。


Finder
File
Edit
View
Go
Window
Help
as Icons ⌘1
✓ as List ⌘2
as Columns ⌘3
as Gallery ⌘4
Use Stacks ^⌘O
Sort By ▶
Clean Up
Clean Up By ▶
Hide Sidebar ⌥⌘S
Hide Preview ⇧⌘P
Hide Toolbar ⌥⌘T
Show All Tabs ⇧⌘\
Hide Tab Bar ⇧⌘T
Hide Path Bar ⌥⌘P
Hide Status Bar ⌘/
Customize Toolbar...
Show View Options ⌘J
Show Preview Options
Enter Full Screen ^⌘F
Search
pre
Menu Items
Preferences...
Hide Preview
Show Preview Options
Show Previous Tab
Help Topics
Use the Preview pane in th...
Change Date & Time prefe...
Change Keyboard Dictatio...
Change iCloud preference...
Set Accessibility preferenc...
Change Displays Options...
Change Security & Privacy...
Change Firewall preferenc...
Change Parental Controls...
Change content caching p...
Show All Help Topics

通过 Search 找快捷键

另外，也可以找一张 macOS 快捷键壁纸当桌面，没事儿时就扫几眼温习一下。想在潜移默化中记住其他工具里的快捷键，设壁纸同样是一个简单有效的办法（更多小技巧可以看这篇文章）。


截屏 Capture the screen
文件管理器 Finder
Mac OS X
常用快捷键一览
切换 Switch
通用 General
浏览器 Safari
系统 System
MacBook

把快捷键设为壁纸，壁纸作者/raner

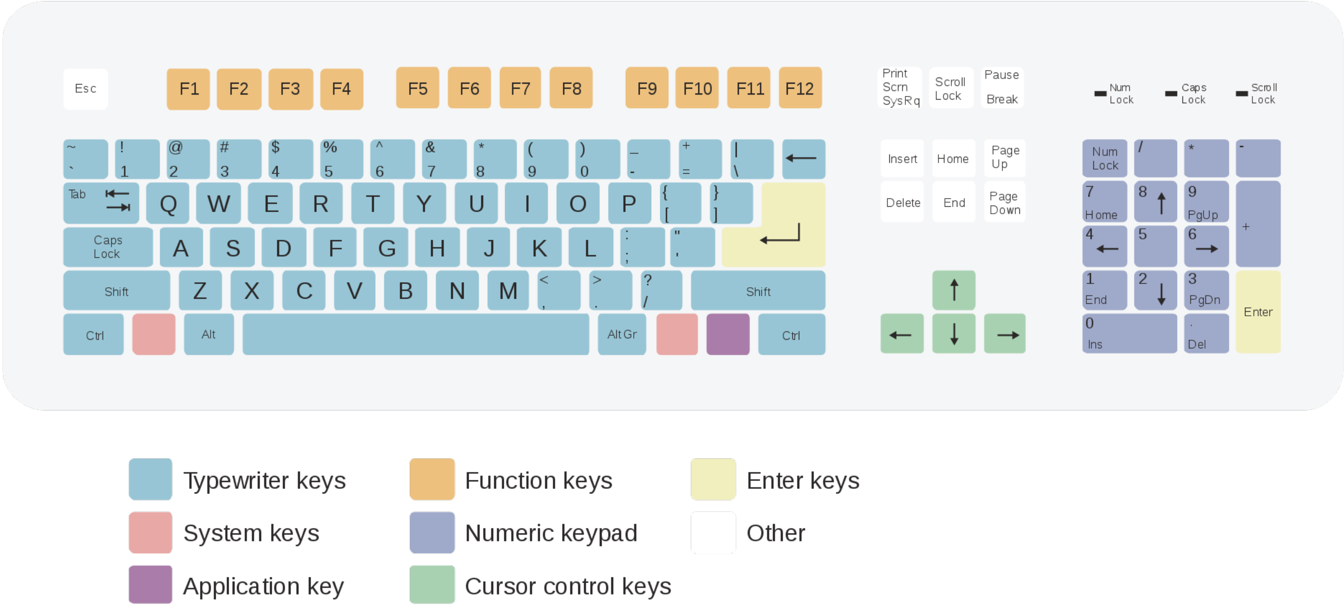
[> 推荐壁纸下载 🔗](#)

Windows 下的系统快捷键

Windows 有 4 颗主要修饰键，微软官方给出了两份清单，分别是《Windows 中的键盘快捷方式》和《应用中的键盘快捷方式》（仅系统自带应用）。

1. `Ctrl`：**控制键**，是 Windows 中最常用的修饰键，可以直接和字母、数字组合形成常用快捷键。是 Control 的缩写，但和 macOS 中的 `^Control` 并不是对应关系。
2. `⎇Alt`：**备用键**，当 Ctrl 键不够用时的补充。也可以用于输入一些特殊字符，比如在小键盘上输入 `Alt` + `0235`，就会出现 ë，相比 macOS 中直接对应特殊字更复杂一些。是 Alternate 的缩写，对应 `⌥Option` 键。

3. `⇧Shift`：**换档键**，对 Ctrl 和 Alt 进行再补充。与上面提到的 ⇧Shift 功能基本一致。
4. `Windows`：**系统键**，默认功能是打开「开始菜单」，与之配合的快捷键常常是与 Windows 有关的操作，比如 `Win + R` 打开运行对话框、`Win + M` 是最小化所有窗口。如果键盘上没有这个键，按下 `Ctrl + Esc` 可以起到相同的作用。

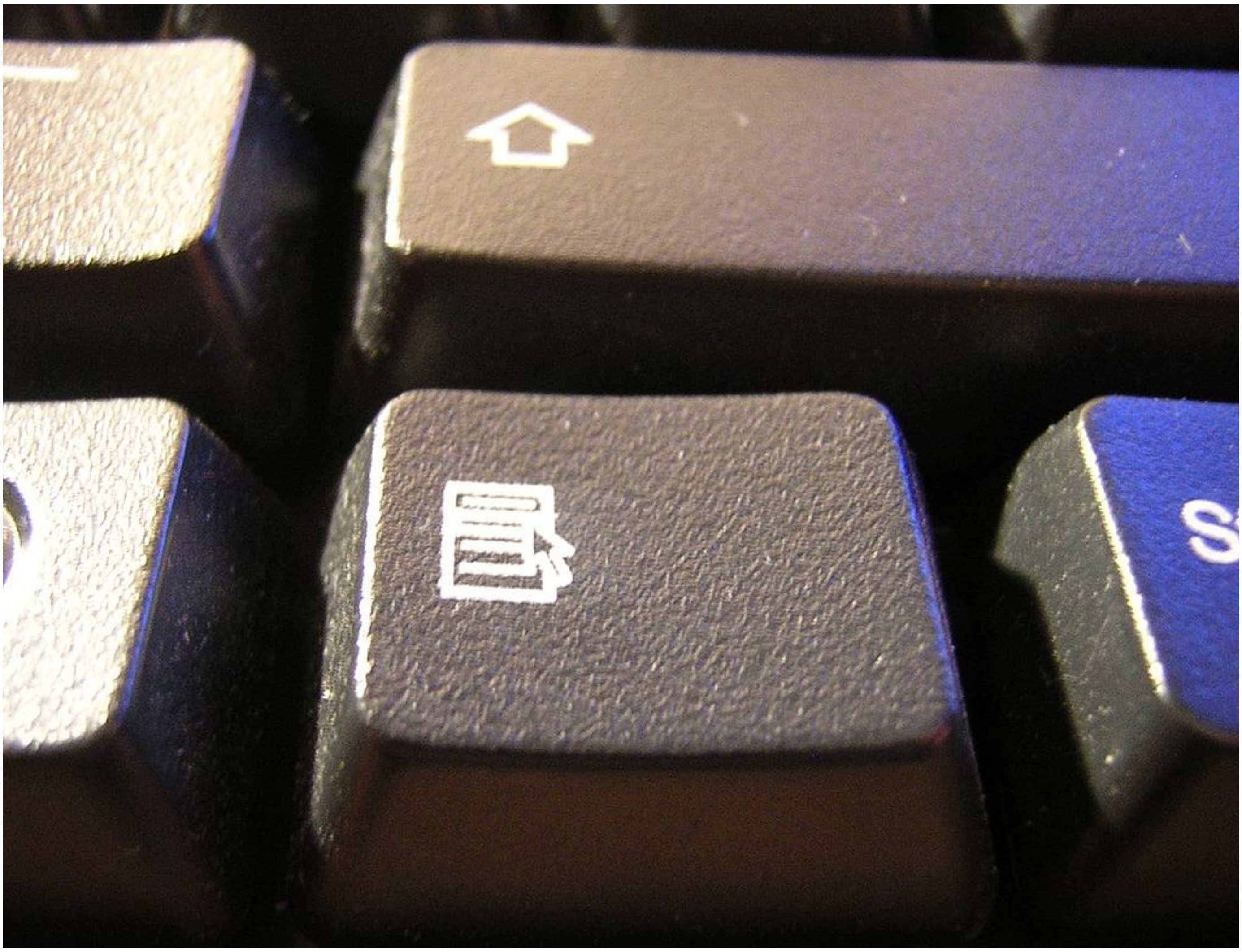

Typewriter keys
Function keys
Enter keys
System keys
Numeric keypad
Other
Application key
Cursor control keys

键盘关系 图/Wikipedia

其实除了上面 4 位，还有一个存在感很低的家伙：`≡Menu` 键。

`≡Menu`：**菜单键**，也称**应用程序键**（上图中的 `Application Key`），通常存在于键盘的右侧，这家伙的存在感非常低，它最初的作用是用来替代**鼠标右键**，所以没有什么组合键与它对应。加之在鼠标和触控板高度普及的今天已经起不到太大作用了，被许多键盘取消也是大势所趋。就连官方也说明

中也没有了它的身影。



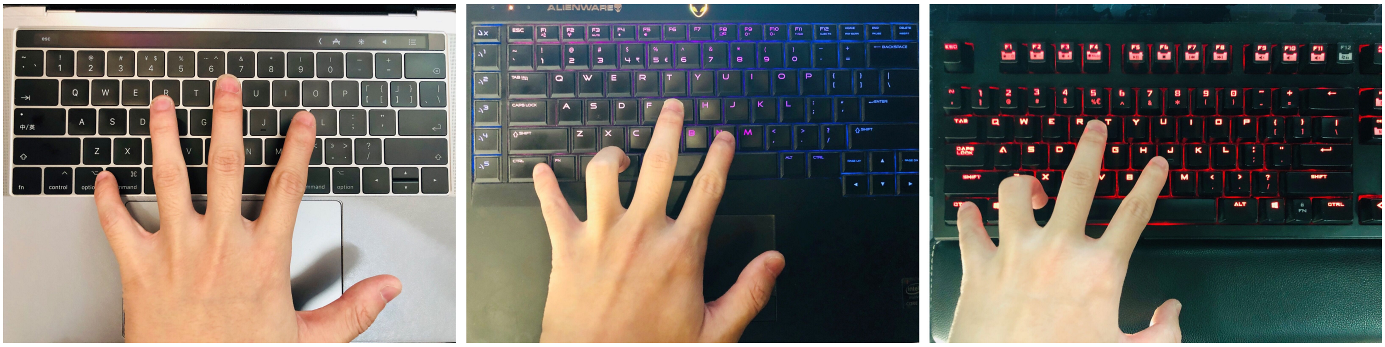
你的键盘上还有 Menu 键吗 图/Wikipedia

Windows 与 macOS 的不同主要在于**键盘的键位和对应关系的不同。**

键位的不同 体现在实际操作上。比如 macOS 下常用的 `⌘Command` 和 `⌥Option` 连在一起，一个小拇指就可以同时按下两个键，从而让整个手掌覆盖大半个键盘，一只手按下大多数快捷键。而 Windows 常用的 `Ctrl` 和 `Alt` 之间通常隔了一个 `Windows` 键，不得不用两个手指分别按下它们。如果组合键较远（比如 Ctrl + Alt + U），用左右侧的组合键去按都很

勉强。还是需要上两只手来操作。

好在系统默认的快捷键很少有这种隔得很远的，当我们自行定义快捷键的时候应当注意这个问题。



Macbook、笔记本、机械键盘布局都有区别

对应关系不同 的问题则会在双系统的用户身上出现。习惯了 Windows 的我初次上手 Mac 时，常常会忘记怎么切换输入法而错误叫出 Spotlight（以至于当时直接把 Spotlight 快捷键换成了别的）。好在苹果贴心的给出了一份 Windows 和 macOS 之间的键盘对应表：《通过“启动转换”在 Windows 中使用 Apple 键盘》。

- `Windows` 徽标键 对应 `⌘Command`
- `⎇Alt` 对应 `⌥Option`
- `⎇Alt`（右侧） 对应 `⌥Option + ^Control`
- `≡Menu` 的说明是「Apple 键盘上没有这个按键」👀。

我们可以从中发现两者之间的区别，`⌘Command` 同时承担了 `Ctrl` 和 `Windows` 键的任务，而`^Control` 键用处较少。这也是为什么同一款软件的双平台快捷键通常会保持对应，但仍然会有一些冲突的原因。

比如在 Windows 下的 Word 替换文本快捷键是 `Ctrl + H`，在 macOS 下 `⌘Command + H` 则是系统级的「隐藏所有窗口」……这些冲突让键盘快捷键很难实现完全对应，需要自己逐渐适应或做一些调整。

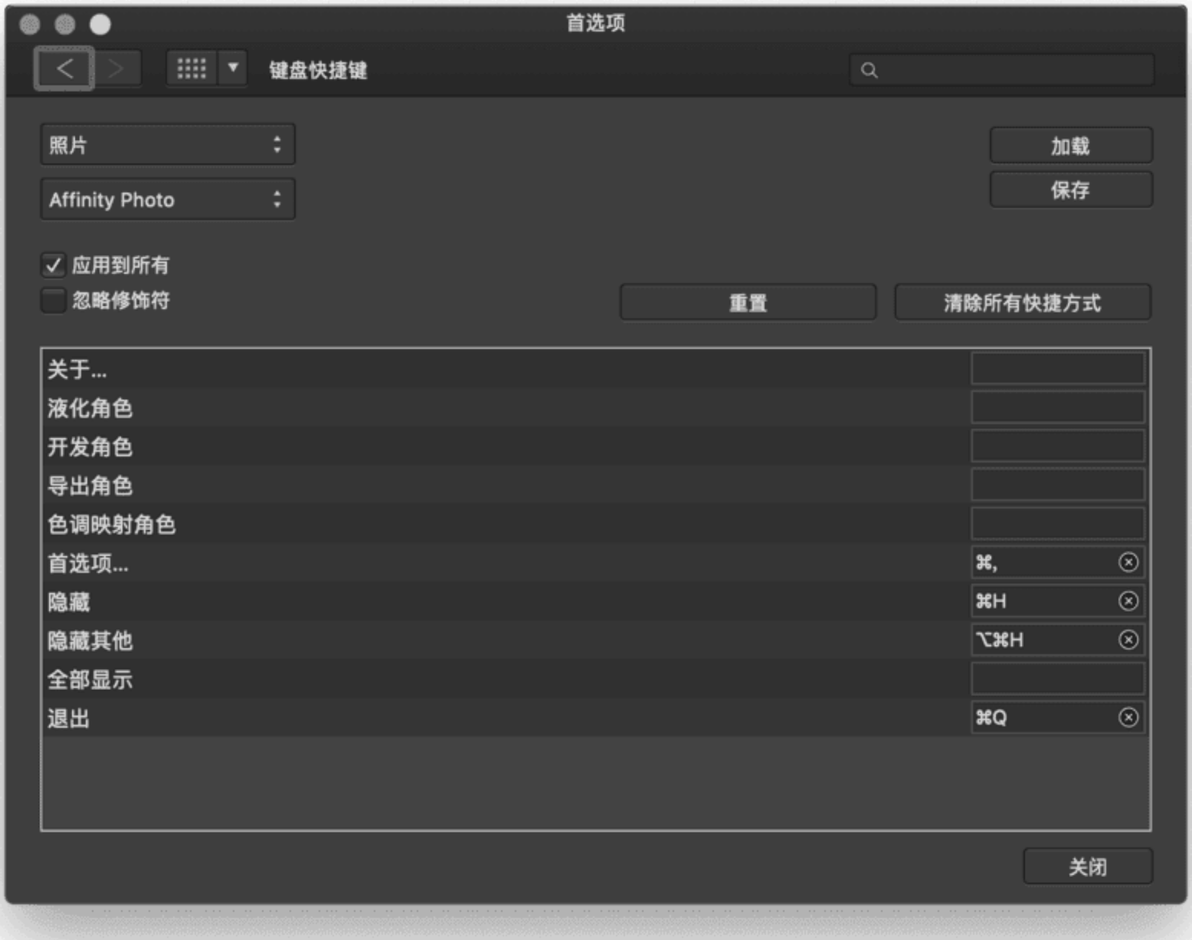
自定义快捷键：以 Hyper Key 为例

当然，并不是每个工具都严格遵守修饰键的原有功能，有些键位设计完全就是胡闹，比如 macOS 的截屏快捷键 `⇧Shift-⌘Command-5`，可能帕格尼尼那拉小提琴的手才能够得上。

根本够不到的键位

即便是 Photoshop、Office 套件等有章可循的工具，同样有

一些键位不符合我们的习惯。不少老道的设计工作者，安装完 Photoshop、Affinity Photo 等等修图工具后，头一件事儿就是去设置里改快捷键。



照片

Affinity Photo

☑ 应用到所有

☐ 忽略修饰符

关于...	
液化角色	
开发角色	
导出角色	
色调映射角色	
首选项...	⌘,
隐藏	⌘H
隐藏其他	⌥⌘H
全部显示	
退出	⌘Q

Affinity Photo 原生支持自定义快捷键

理解，是基础；而当你意识到现有快捷键确实用不习惯，就是时候来自定义快捷键了。

自定义快捷键的方案数不胜数，事实上，在技术上任何按键都是可以自己配置的，你甚至能够把系统快捷键改掉，换成自己喜欢的样子——但这可不是好主意，就像上一节说的，很多预设快捷键有自己的设计道理，改完一时痛快，背地留

有后患[2]

。

所以要把消失已久的 Hyper 键带回来。

这一节，我们将从最基础的系统设置开始，将 Hyper 键带到 macOS 上；另外，同样是 Hyper 键，不同场景下也有进一步、更具体的需求，所以下文还会引入 Karabiner-Elements 、Keyboard Maestro 等第三方工具来进行更深度的自定义。

用原生方法实现 Hyper 键

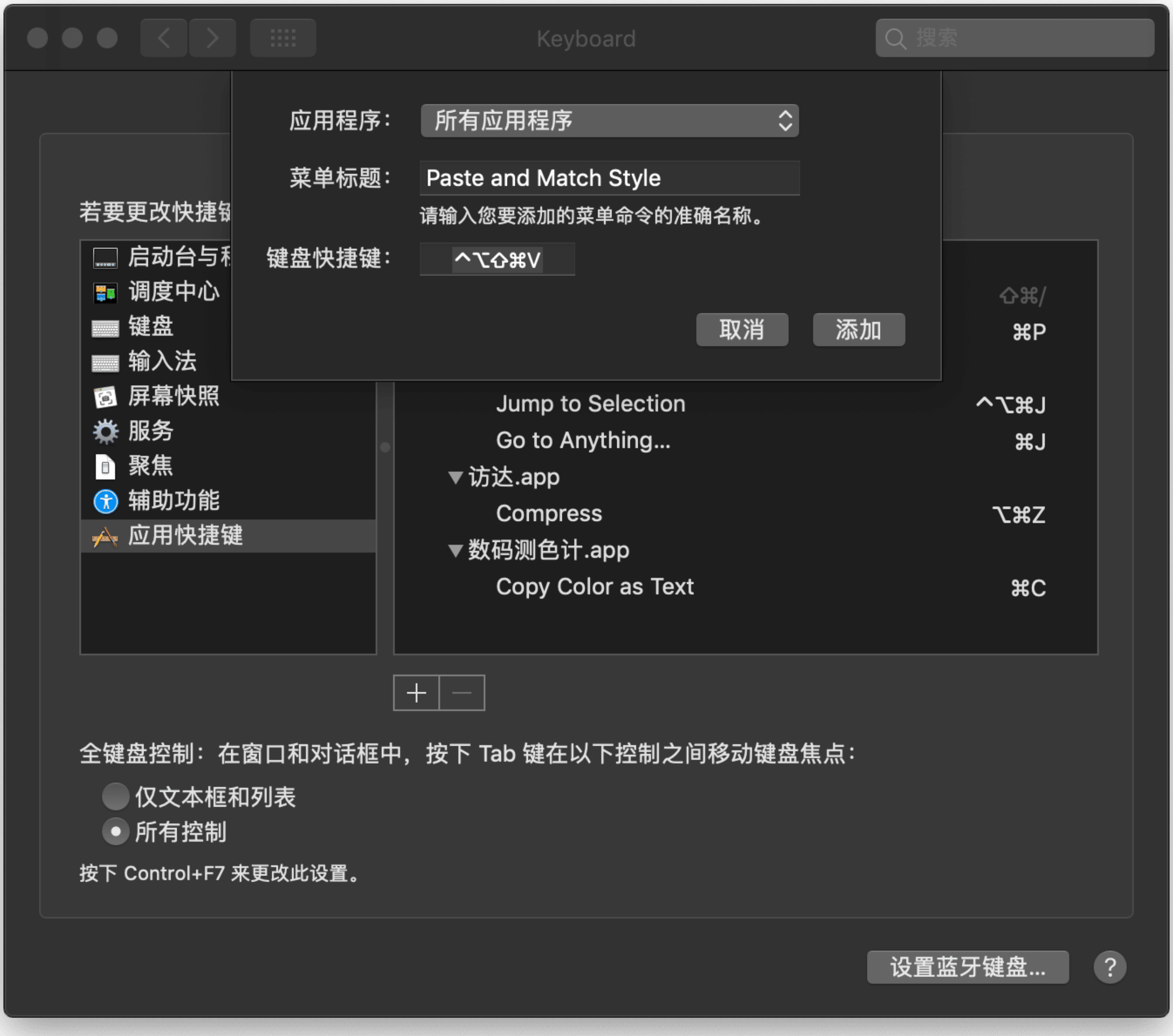
我最喜欢的 Hyper 键方案，是把 `⇧Shift-^Control-⌥Option-⌘Command` 4 颗修饰键一起用上，当成一颗 Hyper 键。这用原生方式也可以实现。

👩‍🏫 小专栏：为什么要用一堆按键来实现 Hyper 键?

1. **避免冲突**：键位非常多，一般不会和预设快捷键重复 (fn)。
2. **容易记忆**：虽然 Hyper 看起来键位多，但可以一股脑把左下角那堆按键全按下去，用起来不假思索。

先来个简单的，在系统键盘设置添加 Hyper 键，实现**快速粘贴纯文本**。其实系统里有对应的快捷键 `⇧Shift-⌥Option-⌘Command-V`，但长长一串很难记，所以还是要引入 Hyper 键。

在「系统设置 - 键盘 - 快捷键 - 应用快捷键」中，添加一个自定义快捷键，填入菜单标题 `Paste and Match Style`，[3] 绑定快捷键 `⇧Shift-^Control-⌥Option-⌘Command-V`。


Keyboard
搜索
若要更改快捷键
启动台与程
调度中心
键盘
输入法
屏幕快照
服务
聚焦
辅助功能
应用快捷键
取消
添加
+
−
仅文本框和列表
所有控制
设置蓝牙键盘…
?

应用程序：所有应用程序

菜单标题：Paste and Match Style

请输入您要添加的菜单命令的准确名称。

键盘快捷键：^⌥⇧⌘V

	⇧⌘/
	⌘P
Jump to Selection	^⌥⌘J
Go to Anything…	⌘J
▼访达.app	
Compress	⌥⌘Z
▼数码测色计.app	
Copy Color as Text	⌘C

全键盘控制：在窗口和对话框中，按下 Tab 键在以下控制之间移动键盘焦点：

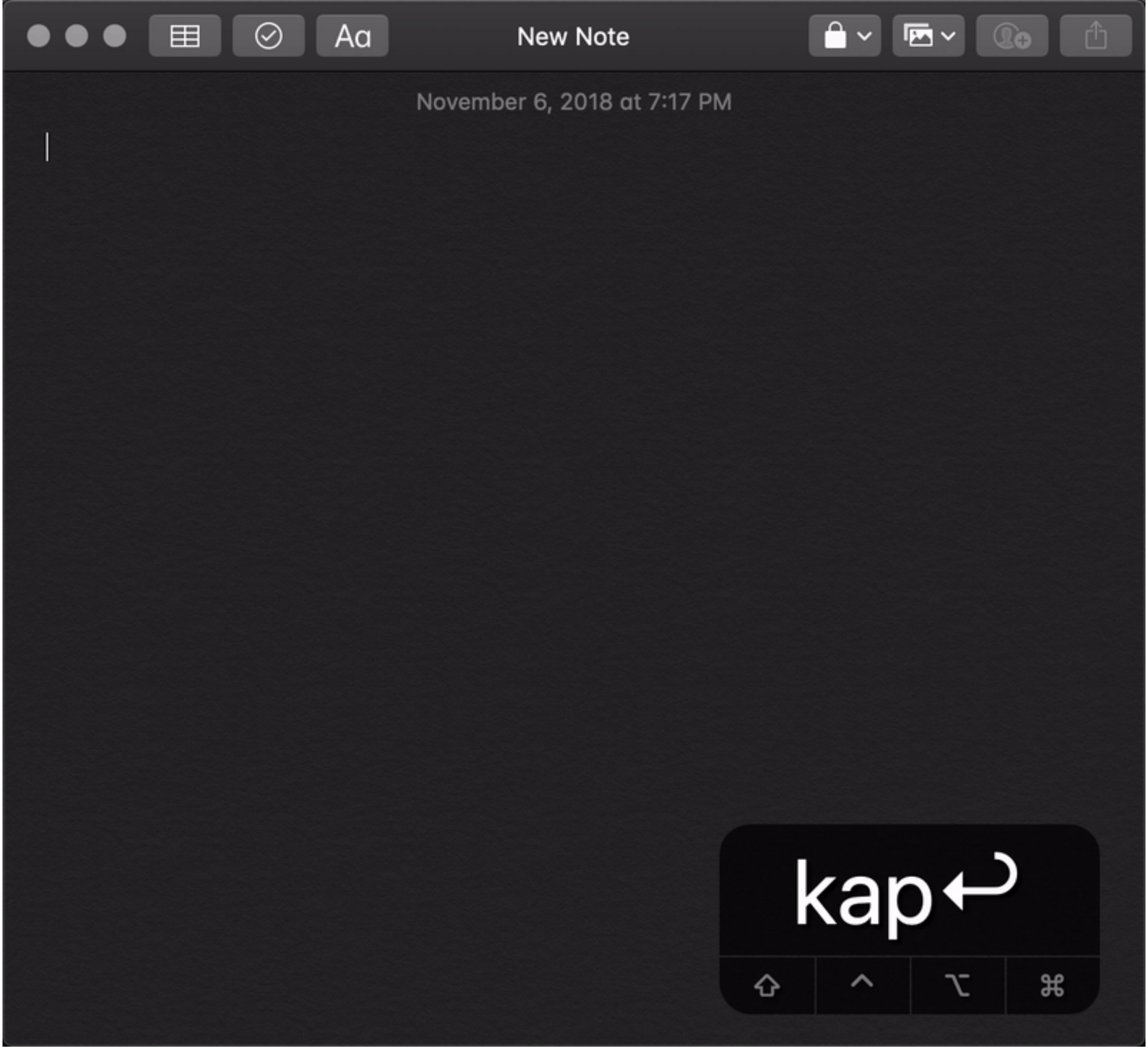
○ 仅文本框和列表

● 所有控制

按下 Control+F7 来更改此设置。

通过系统设置添加 Hyper 键

现在从网页上随意复制一段富文本（带有格式的文本），用刚刚创建的 Hyper 键把它粘到输入界面，就能够得到纯文本内容了。有些记不住系统快捷键的人，可能会把富文本粘贴到 Spotlight 搜索框，再复制下来，兜一圈来获得纯文本，现在有了易记易用的 Hyper 键，粘贴纯文本就省力多了。


New Note
November 6, 2018 at 7:17 PM
kap↩
⇧ ^ ⌥ ⌘

用 Hyper 键粘贴纯文本

想制作其他 Hyper 键，也可以如法炮制。

用 Karabiner-Elements 找回真正的 Hyper 键

本节代码部分改编自 @Nanana 的文章[《macOS 改键利器：Karabiner-Elements 使用详解》](#)，根据我对 Hyper 键的理解有所改动。

通过把一系列修饰键合成一枚 Hyper 键，我们初步找到了一

个较好的自定义快捷键方案。但随着自己制作的 Hyper 键越来越多、使用越来越频繁，一个问题自然而然就出现了：每次要按下那么多键，还是有点不方便[4]

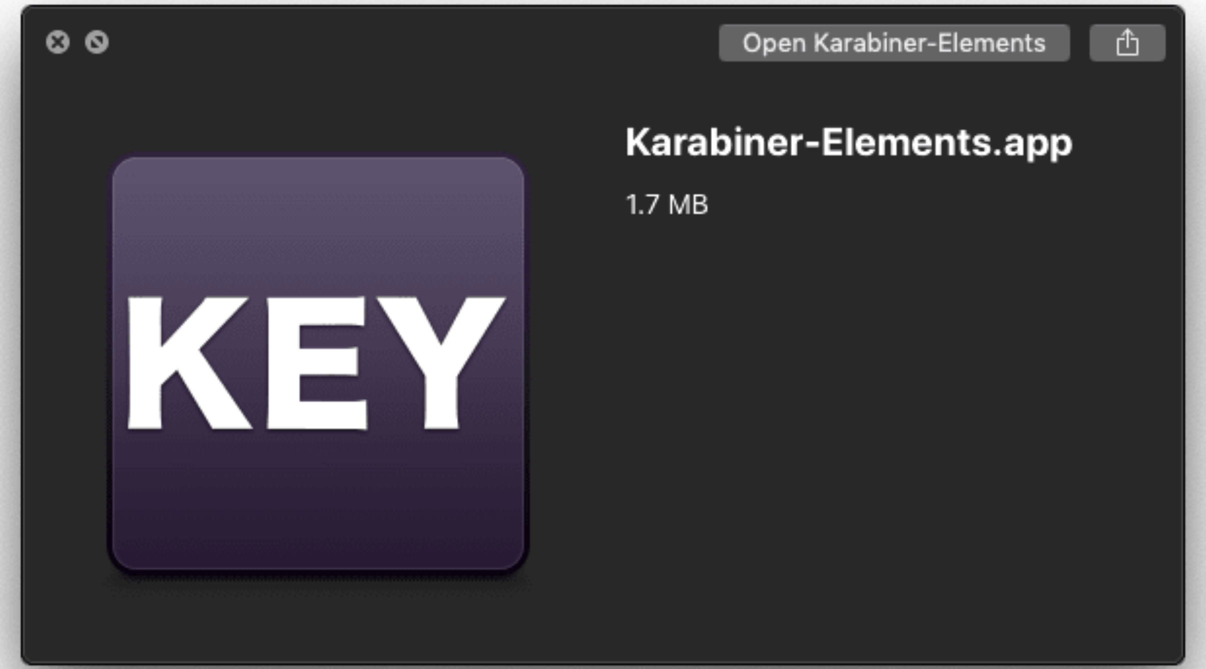
。

所以进一步地，可以借助第三方工具把一堆修饰键「集中」在一颗键上。这样的单颗按键，用起来更接近 Space Cadet 键盘上那颗独立的 Hyper 键。

用单颗 ⌘Command 键模拟 Hyper 键

一般我们会选不常用的按键来充当 Hyper，比如右侧的 `⇧Shift`、`⌘Command` 或 `⌥Option`，也有人更喜欢给 `⇪Caps lock` 键动手术。我们以改动右 `⌘Command` 键为例，由于需要区分左右，得借助 Karabiner-Elements 这个万能改键工具。注意和 Karabiner 区分，后者用于 macOS 10.12 之前的系统，还在用老系统的读者可以参考 @潜水的科技猫 的文章《Control + Option + Shift + Command：带你玩转 macOS

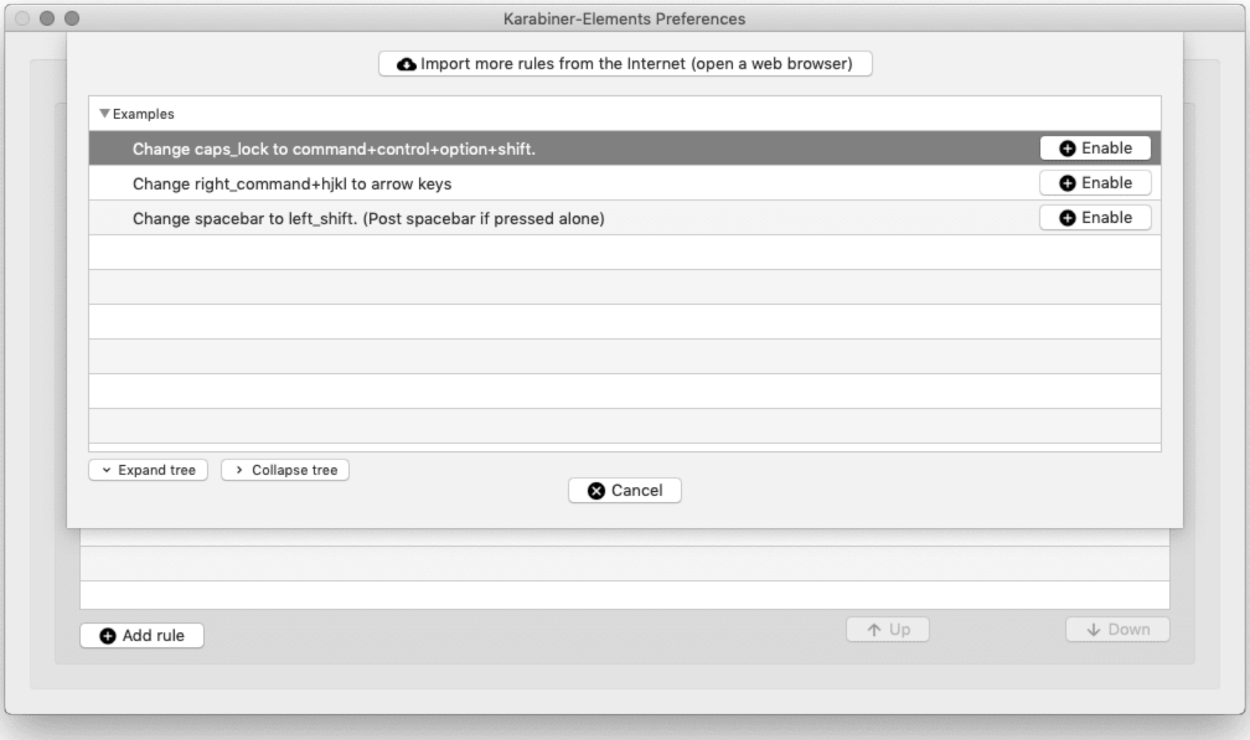
的修饰键》。


Open Karabiner-Elements
KEY
Karabiner-Elements.app
1.7 MB

Karabiner-Elements

回到 Karabiner-Elements 上来。我们的目的很单纯：**把右 ⌘Command 改成 Hyper 键**。别的不管，直奔「Complex Modification」，然后导入（Import）第一个现成的配置[5]

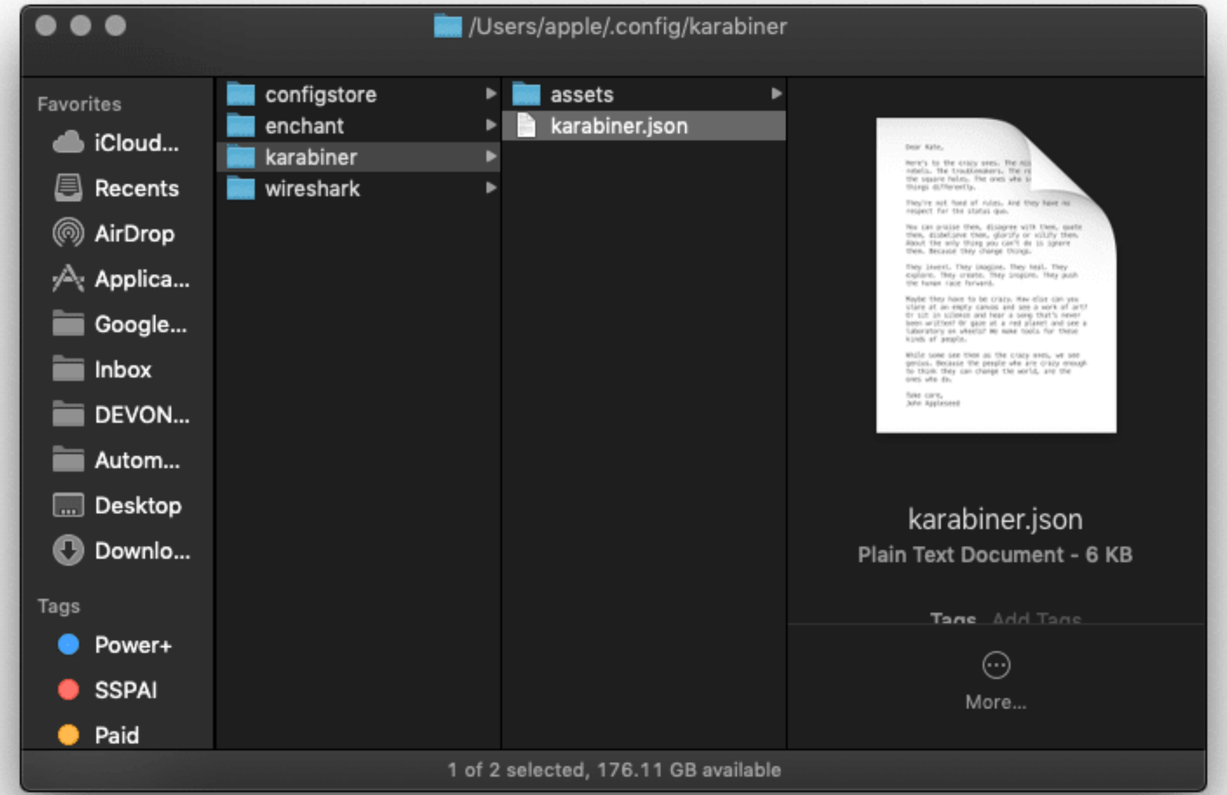
。


Karabiner-Elements Preferences
Import more rules from the Internet (open a web browser)
Examples
Enable
Enable
Enable
Expand tree
Collapse tree
Cancel
Add rule
Up
Down

Change caps_lock to command+control+option+shift.
Change right_command+hjkl to arrow keys
Change spacebar to left_shift. (Post spacebar if pressed alone)

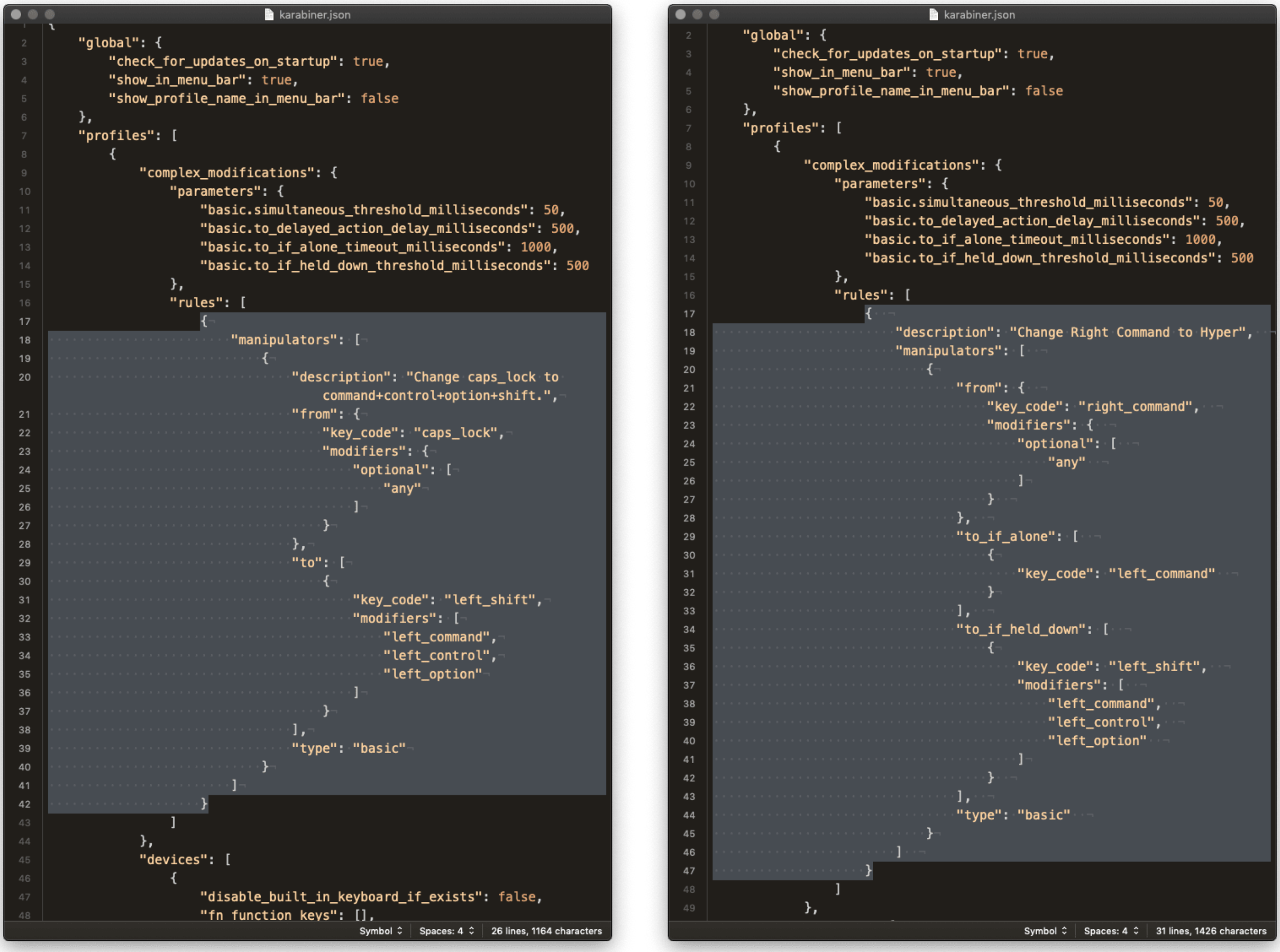
Karabiner-Elements 导入设置

不错，导入的配置是把 `⇪Caps lock` 改成 `⌘Command`，跟我们的目标只有一点点距离了。如果你本来就打算干掉 `⇪Caps lock` 键，这小节下面的部分可以先跳过。打开 Finder，按下快捷键 `⇧Shift-⌘Command-G` 转到 `~/.config/karabiner/karabiner.json`，修改这份配置文件。


/Users/apple/.config/karabiner
Favorites
iCloud...
Recents
AirDrop
Applica...
Google...
Inbox
DEVON...
Autom...
Desktop
Downlo...
Tags
Power+
SSPAI
Paid
configstore
enchant
karabiner
wireshark
assets
karabiner.json
karabiner.json
Plain Text Document - 6 KB
Tags Add Tags
More...
1 of 2 selected, 176.11 GB available

Karabiner-Elements 的配置文件

可以用文本编辑器或者其他你喜欢的编辑器打开它，把稍后的代码替换进下图位置。不是很懂代码或不想折腾的话，可以直接下载 这份 配置文件替换进去。



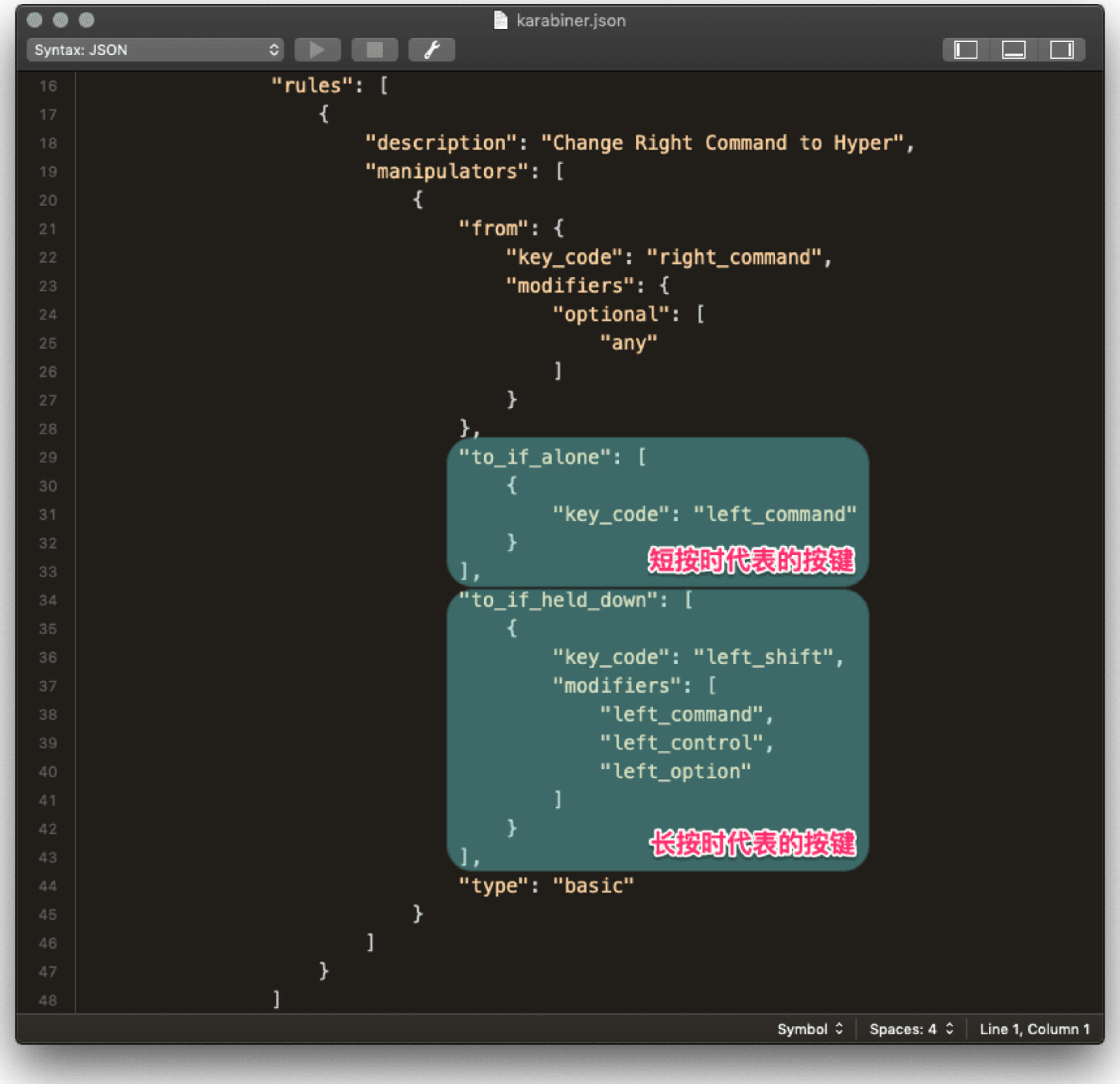
```
    "global": {
        "check_for_updates_on_startup": true,
        "show_in_menu_bar": true,
        "show_profile_name_in_menu_bar": false
    },
    "profiles": [
        {
            "complex_modifications": {
                "parameters": {
                    "basic.simultaneous_threshold_milliseconds": 50,
                    "basic.to_delayed_action_delay_milliseconds": 500,
                    "basic.to_if_alone_timeout_milliseconds": 1000,
                    "basic.to_if_held_down_threshold_milliseconds": 500
                },
                "rules": [
                    {
                        "manipulators": [
                            {
                                "description": "Change caps_lock to command+control+option+shift.",
                                "from": {
                                    "key_code": "caps_lock",
                                    "modifiers": {
                                        "optional": [
                                            "any"
                                        ]
                                    }
                                },
                                "to": [
                                    {
                                        "key_code": "left_shift",
                                        "modifiers": [
                                            "left_command",
                                            "left_control",
                                            "left_option"
                                        ]
                                    }
                                ],
                                "type": "basic"
                            }
                        ]
                    }
                ]
            },
            "devices": [
                {
                    "disable_built_in_keyboard_if_exists": false,
                    "fn_function_keys": [],
```

```
    "global": {
        "check_for_updates_on_startup": true,
        "show_in_menu_bar": true,
        "show_profile_name_in_menu_bar": false
    },
    "profiles": [
        {
            "complex_modifications": {
                "parameters": {
                    "basic.simultaneous_threshold_milliseconds": 50,
                    "basic.to_delayed_action_delay_milliseconds": 500,
                    "basic.to_if_alone_timeout_milliseconds": 1000,
                    "basic.to_if_held_down_threshold_milliseconds": 500
                },
                "rules": [
                    {
                        "description": "Change Right Command to Hyper",
                        "manipulators": [
                            {
                                "from": {
                                    "key_code": "right_command",
                                    "modifiers": {
                                        "optional": [
                                            "any"
                                        ]
                                    }
                                },
                                "to_if_alone": [
                                    {
                                        "key_code": "left_command"
                                    }
                                ],
                                "to_if_held_down": [
                                    {
                                        "key_code": "left_shift",
                                        "modifiers": [
                                            "left_command",
                                            "left_control",
                                            "left_option"
                                        ]
                                    }
                                ],
                                "type": "basic"
                            }
                        ]
                    }
                ]
            },
```

代码替换前后

```
{
    "description": "Change Right Command to Hyper",
    "manipulators": [
        {
            "from": {
                "key_code": "right_command",
                "modifiers": {
                    "optional": [
                        "any"
                    ]
                }
            },
            "to_if_alone": [
                {
                    "key_code": "left_command"
                }
            ],
            "to_if_held_down": [
                {
                    "key_code": "left_shift",
                    "modifiers": [
                        "left_command",
                        "left_control",
                        "left_option"
                    ]
                }
            ],
            "type": "basic"
        }
    ]
}
```

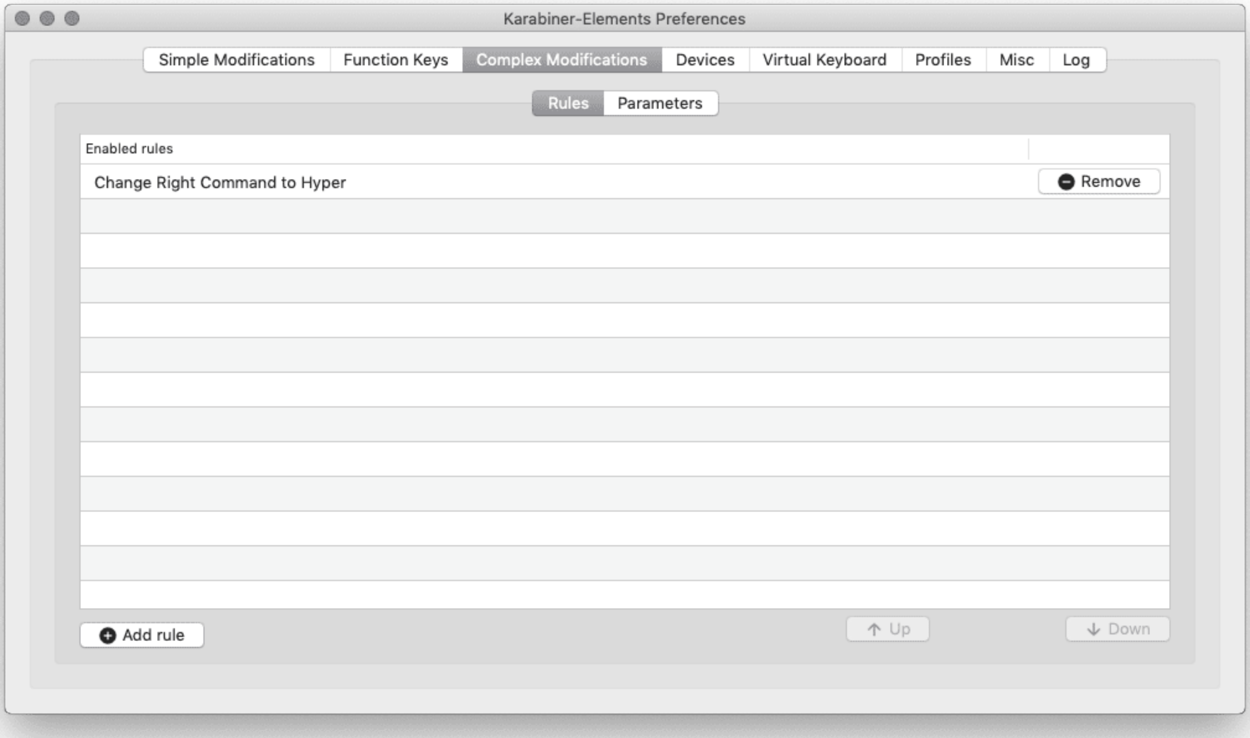
如果不想改动 ⌘Command 键，可以把 `right_command` 换成你想要的其他键，比如 `right_option`，同时记得 `"key_code": "left_command"` 也要改成 `"key_code": "left_option"`，这表示**短按下去时仍然把右 ⌥Option 当 ⌥Option 用，只在稍稍长按时才用作 Hyper 键**。更详细的配置，推荐各位读读 @Nanana 的 原文，他已经写得非常详尽了。


karabiner.json
Syntax: JSON
短按时代表的按键
长按时代表的按键
Symbol
Spaces: 4
Line 1, Column 1

```
                "rules": [
                    {
                        "description": "Change Right Command to Hyper",
                        "manipulators": [
                            {
                                "from": {
                                    "key_code": "right_command",
                                    "modifiers": {
                                        "optional": [
                                            "any"
                                        ]
                                    }
                                },
                                "to_if_alone": [
                                    {
                                        "key_code": "left_command"
                                    }
                                ],
                                "to_if_held_down": [
                                    {
                                        "key_code": "left_shift",
                                        "modifiers": [
                                            "left_command",
                                            "left_control",
                                            "left_option"
                                        ]
                                    }
                                ],
                                "type": "basic"
                            }
                        ]
                    }
                ]
```

Karabiner-Elements 代码简析

保存好修改后的配置，就能看到新的 Hyper 键出现在了 Karabiner-Elements 设置界面中。


Karabiner-Elements Preferences
Simple Modifications
Function Keys
Complex Modifications
Devices
Virtual Keyboard
Profiles
Misc
Log
Rules
Parameters
Remove
Add rule
Up
Down

Enabled rules

Change Right Command to Hyper

Karabiner-Elements 中设置好后的 Hyper 键

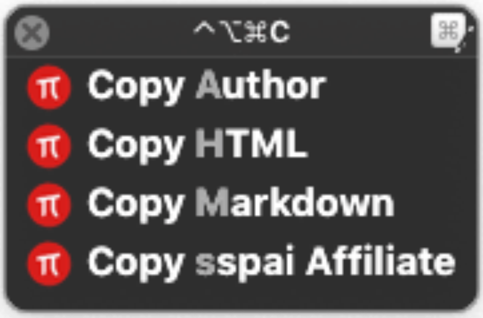
如果你还保留着上一节的示例 Hyper 设置，现在找个编辑器来按一下 右 ⌘Command-V，是不是能够直接粘贴纯文本了？当然录屏时显示的还是按下 4 颗修饰键，所以下面直接放拍摄的效果：

用右侧 ⌘Command 代替一堆修饰键，完成 Hyper 键效果

搞定。

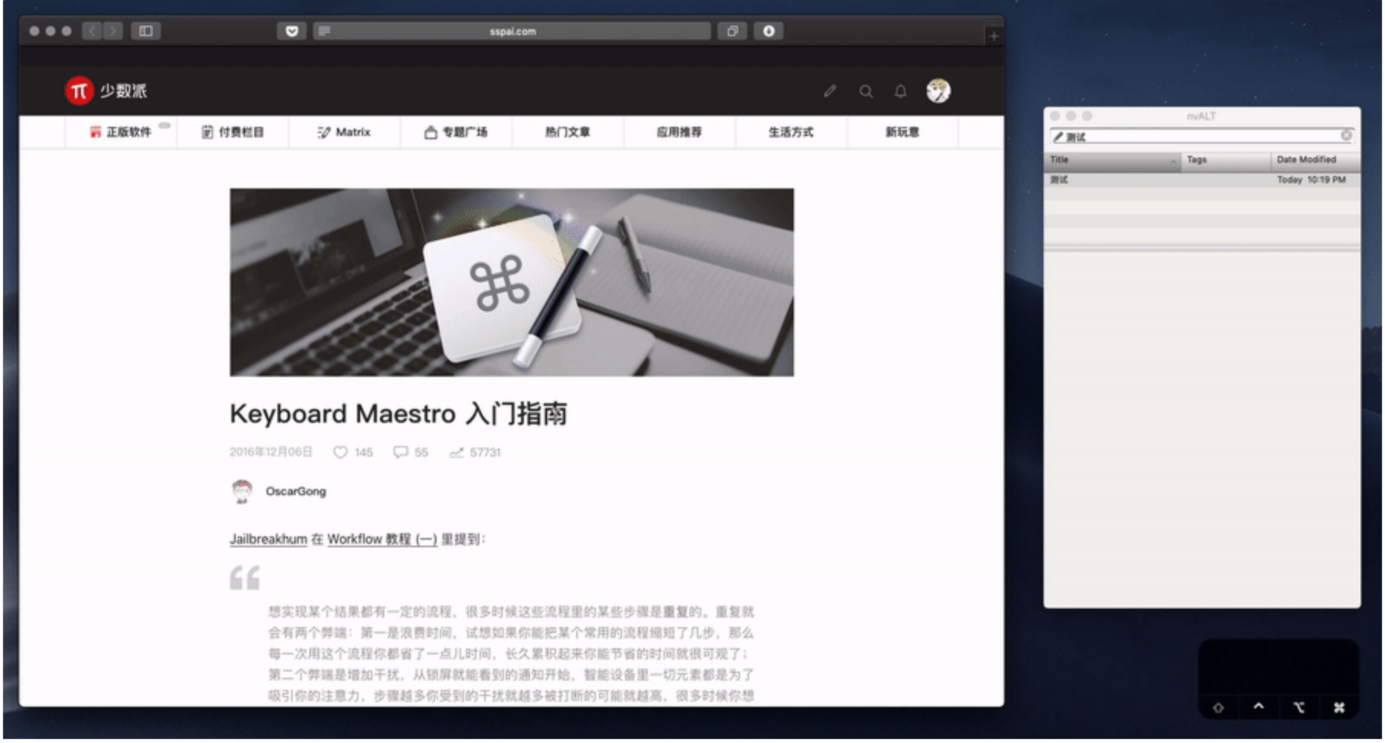
用 Keyboard Maestro 把同类功能统合起来

Hyper 键设置多了之后，不小心还是会打架，经常发生重复。**Keyboard Maestro 的 Palette 功能是解决快捷键冲突最聪明的方式之一。**


^⌥⌘C
Copy Author
Copy HTML
Copy Markdown
Copy sspai Affiliate

Palette 小浮窗

首先你当然要把 Hyper 键和对应的操作搬进 Keyboard Maestro。当遇上快捷键冲突时，Palette——一个显示冲突快捷键的小悬浮窗——就会弹出。根据 Palette 中动作名的颜色提示，你可以紧跟着按下字母/数字键来触发 Macro（自动化动作）。如图，我按下 Hyper 键后弹出了一个汇集了各种拷贝操作的 Palette，再按 `M`，就能拷贝当前标签页链接为 Markdown 格式。


sspai.com
少数派
正版软件
付费栏目
Matrix
专题广场
热门文章
应用推荐
生活方式
新玩意
nvALT
测试
Title
Tags
Date Modified
测试
Today 10:19 PM

Keyboard Maestro 入门指南

2016年12月06日 ♡ 145 💬 55 57731

OscarGong

Jailbreakhum 在 Workflow 教程 (一) 里提到：

> 想实现某个结果都有一定的流程，很多时候这些流程里的某些步骤是重复的。重复就会有两个弊端：第一是浪费时间，试想如果你能把某个常用的流程缩短了几步，那么每一次用这个流程你都省了一点儿时间，长久累积起来你能节省的时间就很可观了；第二个弊端是增加干扰，从锁屏就能看到的通知开始，智能设备里一切元素都是为了吸引你的注意力，步骤越多你受到的干扰就越多被打断的可能就越高，很多时候你想

用 Palette 把相关功能的 Hyper 键整合起来

有了 Keyboard Maestro，就不用绞尽脑汁设计不容易冲突的 Hyper 键。唯一需要注意的，就是尽量用**英文**来命名 Macro，不然弹出 Palette 后没法通过键盘定位，还得那鼠标磨蹭半天。

关于 Keyboard Maestro 中的 Palette，@潜水的科技猫、@Oscar_Gong 以及 @帕特里柯基 都是高手，大家可以读读他们的文章。

拓展阅读：

- 《Keyboard Maestro 入门指南》
- 《Macro Palette 让我耳目一新的交互方式》
- 《A Modern Space Cadet / Steve Losh》

👩‍🏫 小专栏：为什么要用第三方改键工具?

有的读者可能会问，既然系统自带改键功能、部分 App 也能修改快捷键，为什么还要用 Karabiner-Elements、Keyboard Maestro 这些三方工具？目的就是要**统一管理**：

1. **统一修改**：可以在一个界面里面统一管理所有的快捷键，免得去各个分散的 App 里设置。
2. **避免冲突**：可以查看所有的自定义快捷键，避免自己和自己打架。
3. **备份方便**：可以备份配置文件，换电脑、刷系统后直接恢复到原来的键位，不用重新设置。

所以，有条件使用第三方工具的读者，也建议把系统中的自定义快捷键迁移过去。

一类特殊的快捷键：Double Key

对于 Double 我没有找到参考资料，算是本人半原创的一类快捷键吧，姑且先把技巧记录在此；如果有读者知道出处或者有和我类似的习惯，欢迎留言评论。

在 Hyper 键之外，我还设置了一类特殊的自定义快捷键：**Double Key（双击键）**，就是各种**双击**，具体的例子有：

- 在 Safari 中，连按两次 `⌘Command-P` 直接打印打印当前页面（详见）
- 在任何应用中，连按两次 `⌘Command-Q` 才能退出应用，避免误触退出。

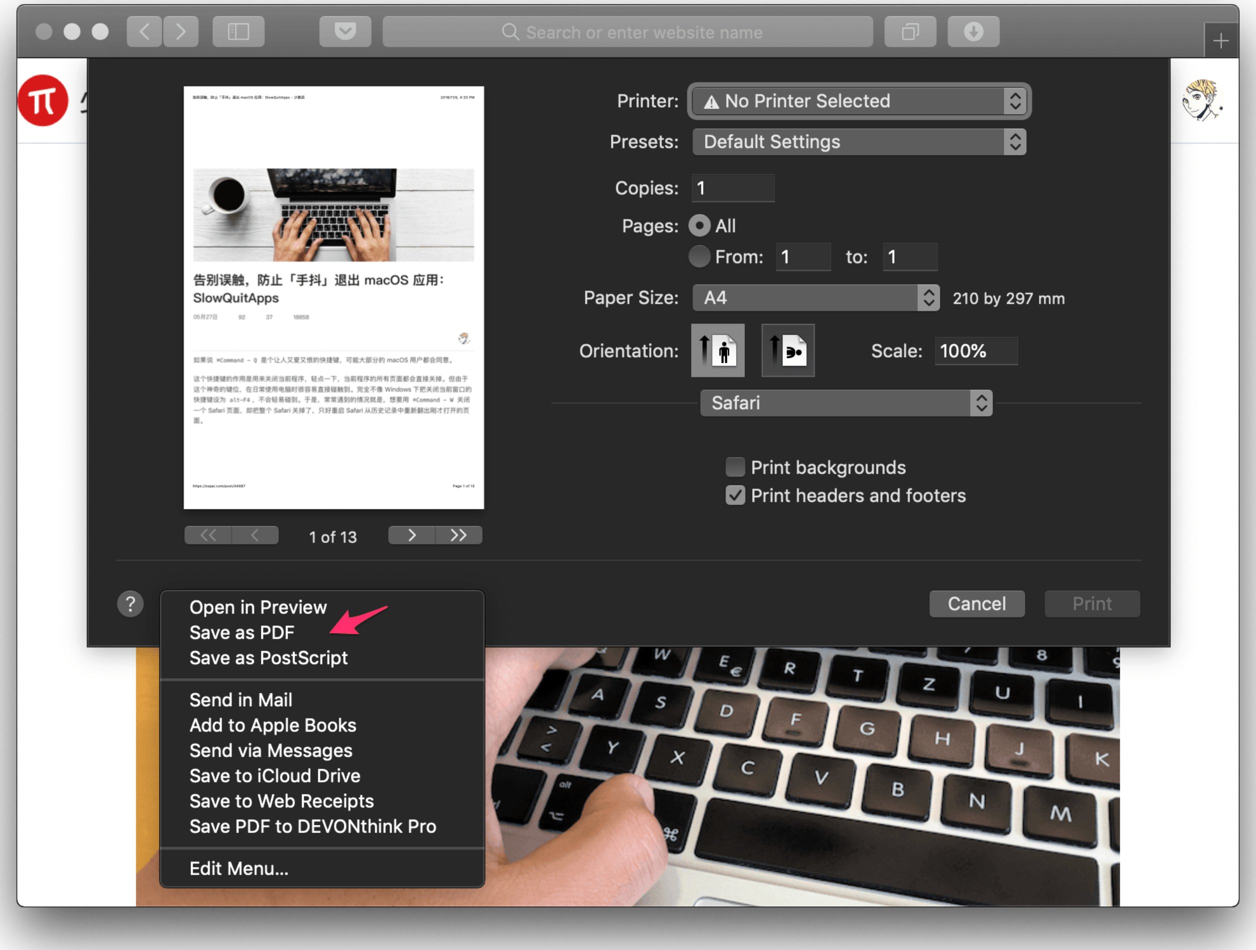
可见，**Double Key 不需另记新的快捷键，只是改变原有功**

能操作起来的难易程度，它既能**简化操作——**把需要鼠标点好几下的操作（比如打印页面）连成一步，也可以**让操作变得更难，以避免误触发生**。

下面就拿我常用的两个 Double Key 来抛砖引玉，希望大家能用得上。

快速打印：双击 ⌘Command-P

在 macOS 上，文档、网页都能保存成 PDF，但是这个「Save as PDF（保存为 PDF）」的按钮却躲在角落的菜单里，鼠标挪过去点很不方便。

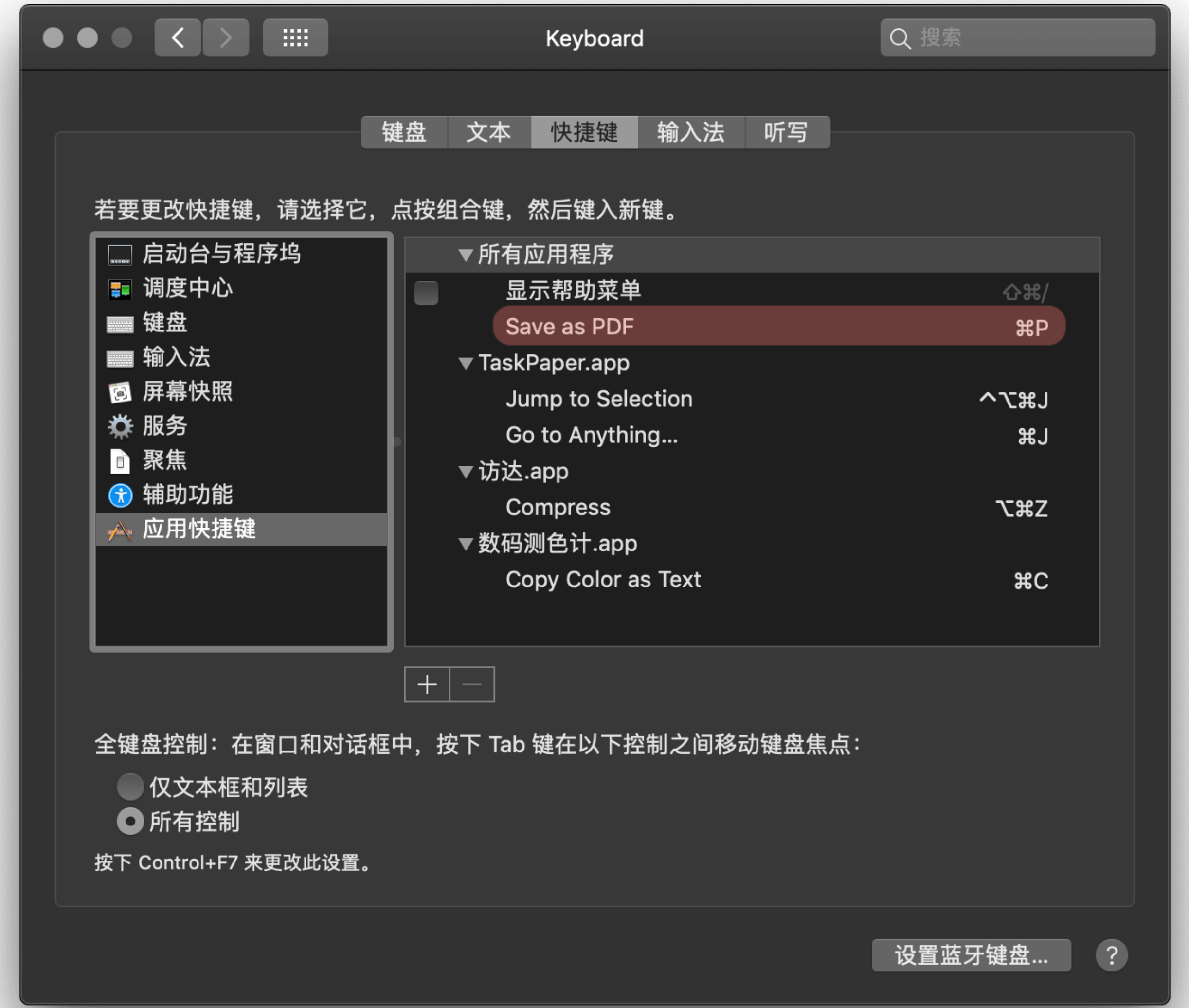


Printer: No Printer Selected
Presets: Default Settings
Copies: 1
Pages: ● All
○ From: 1 to: 1
Paper Size: A4 210 by 297 mm
Orientation:
Scale: 100%
Safari
☐ Print backgrounds
☑ Print headers and footers

Search or enter website name
告别误触，防止「手抖」退出 macOS 应用：SlowQuitApps
1 of 13
?
Open in Preview
Save as PDF
Save as PostScript
Send in Mail
Add to Apple Books
Send via Messages
Save to iCloud Drive
Save to Web Receipts
Save PDF to DEVONthink Pro
Edit Menu...
Cancel
Print

保存 PDF 的按钮藏得太深

考虑到「Save as PDF」其实是一个菜单栏按钮，可以在「系统设置 - 键盘 - 快捷键 - 应用快捷键」中为它绑上一个快捷键来简化操作。我的 电脑是英文系统，所以填了「Save as PDF」，中文系统下应该是「存储为 PDF」。


Keyboard

搜索

键盘 文本 快捷键 输入法 听写

若要更改快捷键，请选择它，点按组合键，然后键入新键。

- 启动台与程序坞
- 调度中心
- 键盘
- 输入法
- 屏幕快照
- 服务
- 聚焦
- 辅助功能
- 应用快捷键

▼所有应用程序	
☐ 显示帮助菜单	⇧⌘/
Save as PDF	⌘P
▼TaskPaper.app	
Jump to Selection	^⌥⌘J
Go to Anything...	⌘J
▼访达.app	
Compress	⌥⌘Z
▼数码测色计.app	
Copy Color as Text	⌘C

+ −

全键盘控制：在窗口和对话框中，按下 Tab 键在以下控制之间移动键盘焦点：

○ 仅文本框和列表
● 所有控制

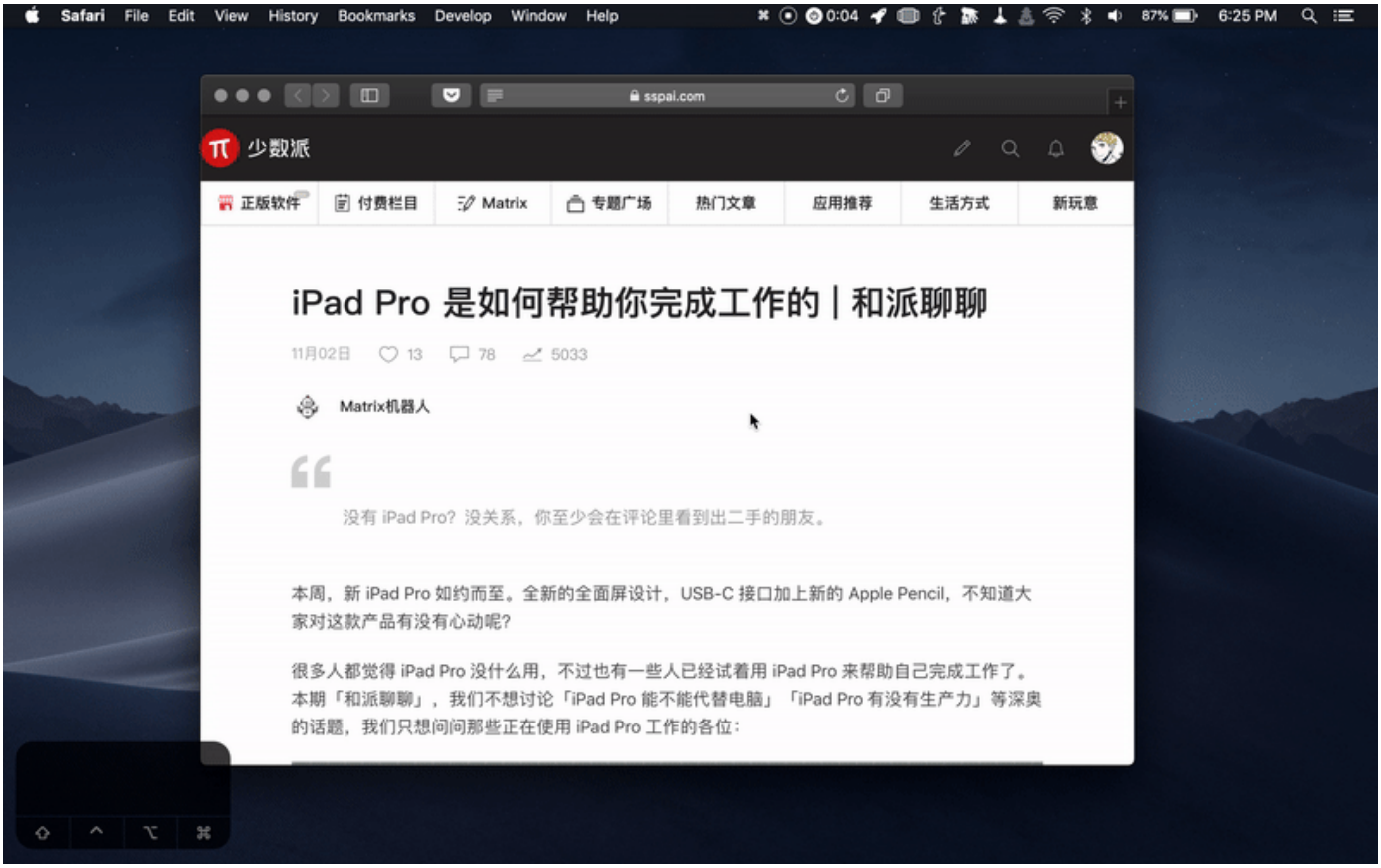
按下 Control+F7 来更改此设置。

设置蓝牙键盘... ?

为保存 PDF 设置系统快捷键

我选择的保存快捷键和打印一样都是 `⌘Command-P`，这样一来，就可以在按下 `⌘Command-P` 展开打印窗口后，继续通过

⌘Command–P 来快速保存。


Safari
File
Edit
View
History
Bookmarks
Develop
Window
Help
0:04
87%
6:25 PM
sspai.com
少数派
正版软件
付费栏目
Matrix
专题广场
热门文章
应用推荐
生活方式
新玩意
11月02日
13
78
5033
Matrix机器人
⇧
^
⌥
⌘

iPad Pro 是如何帮助你完成工作的｜和派聊聊

没有 iPad Pro？没关系，你至少会在评论里看到出二手的朋友。

本周，新 iPad Pro 如约而至。全新的全面屏设计，USB-C 接口加上新的 Apple Pencil，不知道大家对这款产品有没有心动呢？

很多人都觉得 iPad Pro 没什么用，不过也有一些人已经试着用 iPad Pro 来帮助自己完成工作了。本期「和派聊聊」，我们不想讨论「iPad Pro 能不能代替电脑」「iPad Pro 有没有生产力」等深奥的话题，我们只想问问那些正在使用 iPad Pro 工作的各位：

双击快速打印

实际上，你可以飞快地连按，系统仍然能够正确识别快捷键。这样就实现了「连按两次 ⌘Command–P 来快速打印」的效果，比专门另外记一个快捷键要更方便。操作起来也更连贯。

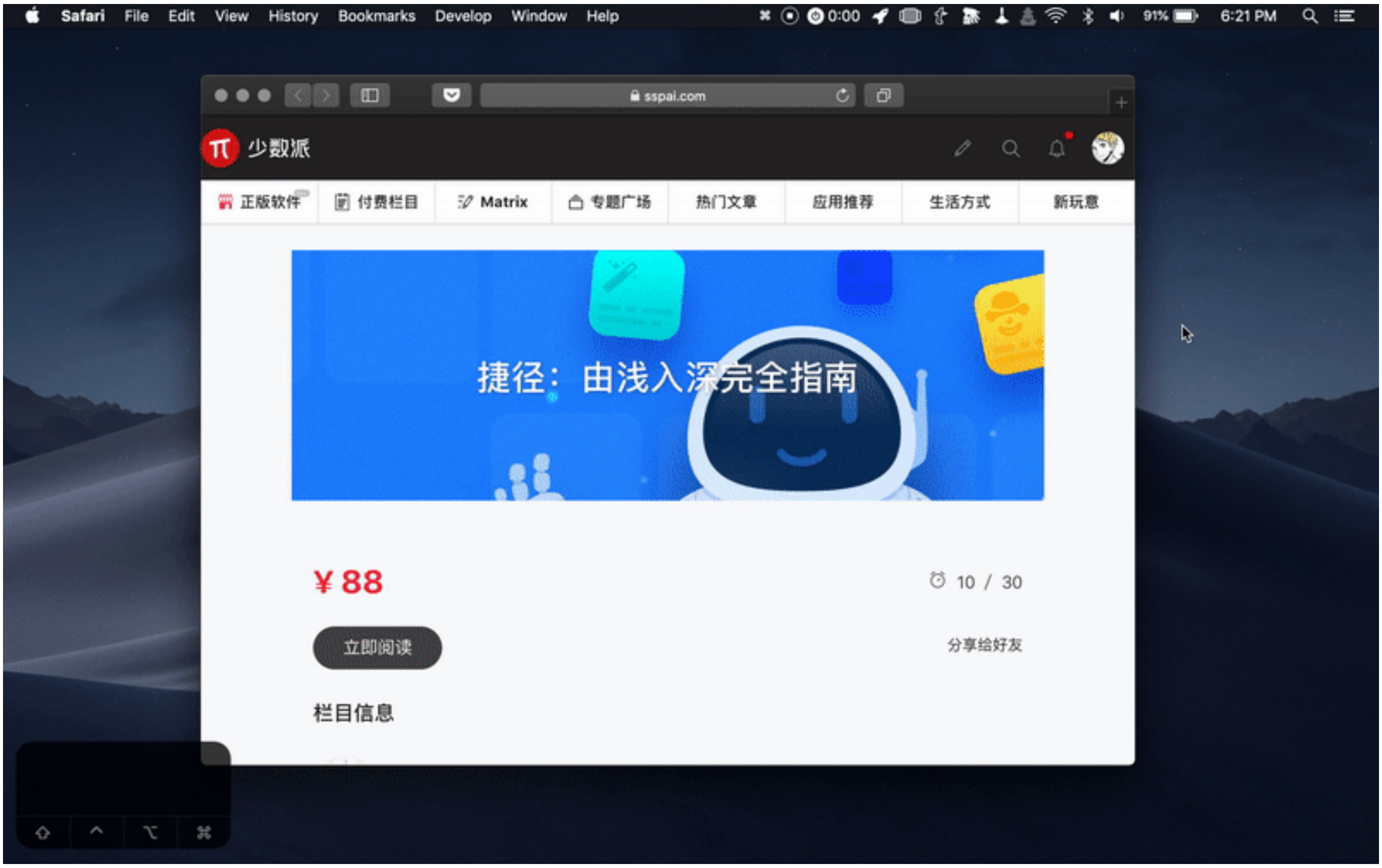
避免误触：双击 ⌘Command-Q

注：本节技巧需要借助 Keyboard Maestro。

⌘Command–Q 这个键位设计不靠谱，想关个网页、切个标签页都很容易按到退出。如果你打算靠 ⌘Command–W 关掉网页，很可能会手抖把整个应用都退出掉，成为段子里「老板走近时

对着电脑桌面傻笑的员工」。

不如用 Double Key 代替单个 ⌘Command-Q，给电脑上一道保险。下面的演示里，需要连按两次 ⌘Command-Q 才能退出应用[6]，既不怕误触，真想退出应用时也可以快速连按。一点点操作难度的提升，带来了更安全的操作。

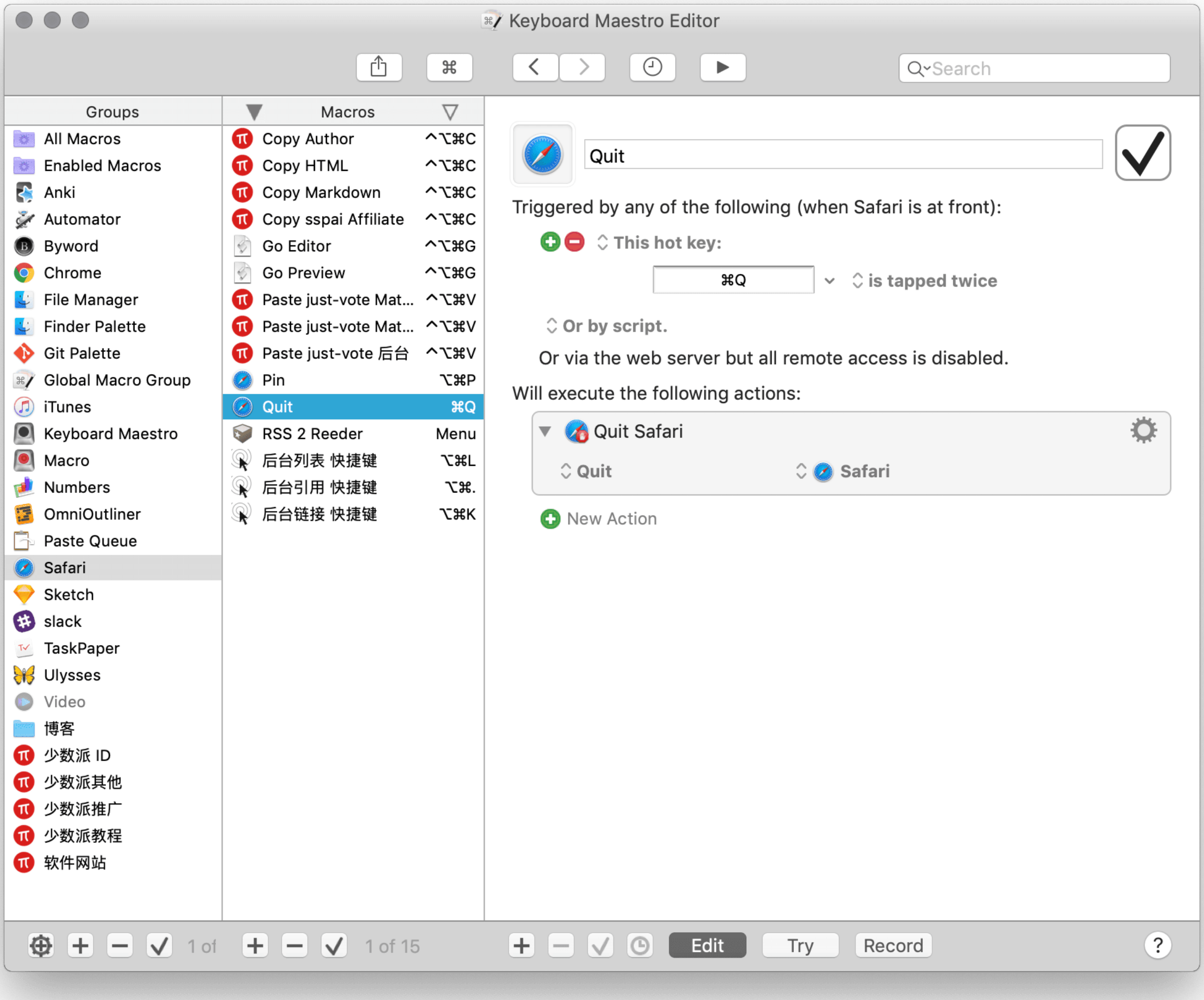

Safari File Edit View History Bookmarks Develop Window Help
0:00
91%
6:21 PM
sspai.com
少数派
正版软件
付费栏目
Matrix
专题广场
热门文章
应用推荐
生活方式
新玩意
捷径：由浅入深完全指南
¥88
10 / 30
立即阅读
分享给好友
栏目信息
⇧ ^ ⌥ ⌘

双击退出 Safari

和快速打印的动作不同，这次双击其实对应的是一个动作（而两次 ⌘Command-P 其实各自负责显示打印窗口和保存文件两个动作），超出了系统设置的能力范围，需要引入第三方工具。

以双击退出 Safari 为例，我们用 Keyboard Maestro 来实现。

可以看到，Keyboard Maestro 自带了**「This hotkey is tapped twice（按两边快捷键）」**这一触发方式，拿来用就是。



Keyboard Maestro Editor

Quit

Triggered by any of the following (when Safari is at front):

This hot key: ⌘Q is tapped twice

Or by script.

Or via the web server but all remote access is disabled.

Will execute the following actions:

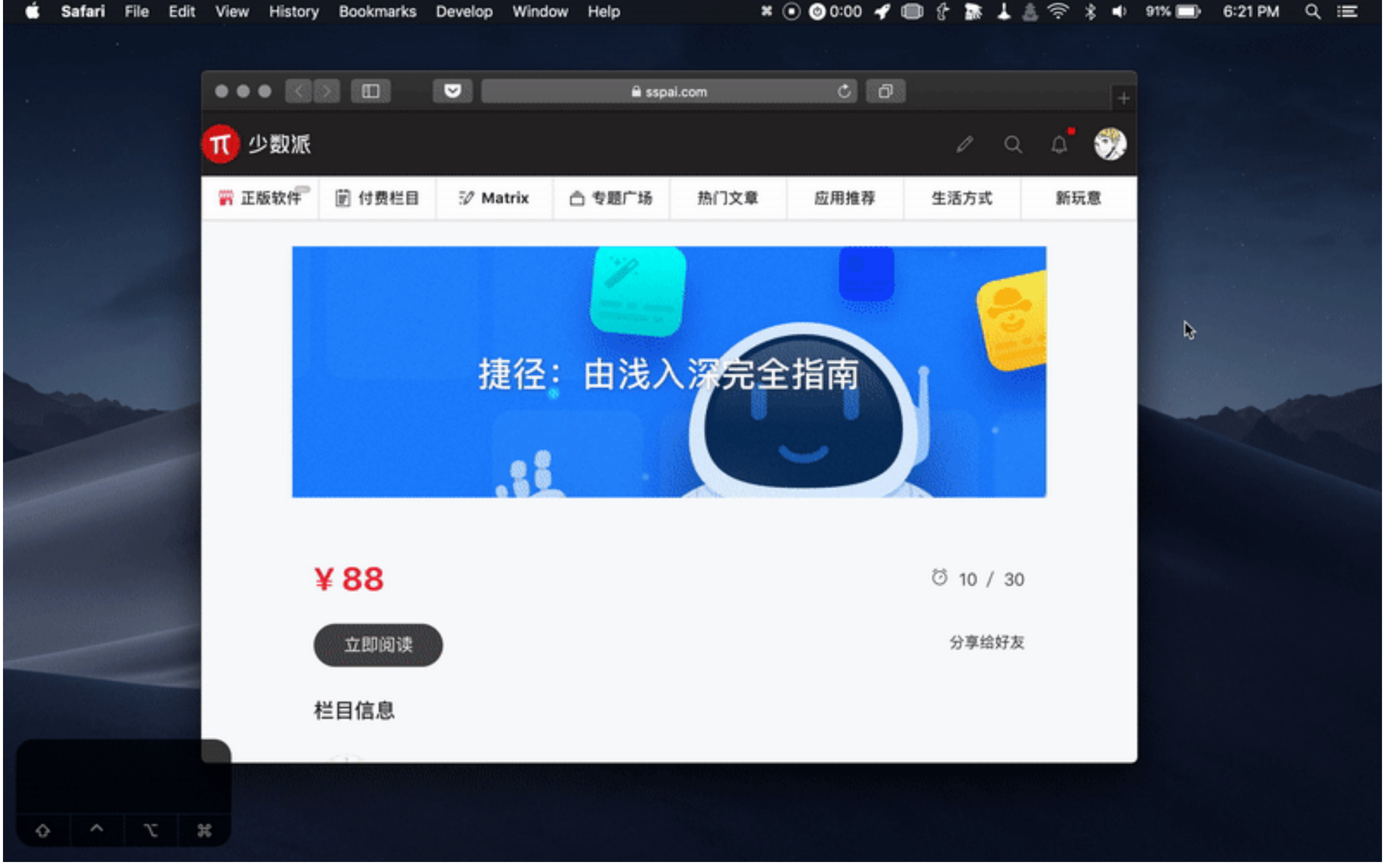
Quit Safari

Quit Safari

Search
Groups
All Macros
Enabled Macros
Anki
Automator
Byword
Chrome
File Manager
Finder Palette
Git Palette
Global Macro Group
iTunes
Keyboard Maestro
Macro
Numbers
OmniOutliner
Paste Queue
Safari
Sketch
slack
TaskPaper
Ulysses
Video
博客
少数派 ID
少数派其他
少数派推广
少数派教程
软件网站
Macros
Copy Author ^⌥⌘C
Copy HTML ^⌥⌘C
Copy Markdown ^⌥⌘C
Copy sspai Affiliate ^⌥⌘C
Go Editor ^⌥⌘G
Go Preview ^⌥⌘G
Paste just-vote Mat... ^⌥⌘V
Paste just-vote Mat... ^⌥⌘V
Paste just-vote 后台 ^⌥⌘V
Pin ⌥⌘P
Quit ⌘Q
RSS 2 Reeder Menu
后台列表 快捷键 ⌥⌘L
后台引用 快捷键 ⌥⌘.
后台链接 快捷键 ⌥⌘K
New Action
1 of
1 of 15
Edit
Try
Record
?

用 Keyboard Maestro 设置 Double Key

接着把对应的动作设为「Quit Safari」，这样系统自带的单击 `⌘Command-Q` 就会被自定义的快捷键「占用」，Double Key 便实现了。打开 Safari，可以看到 Double Key 已经生效。


Safari File Edit View History Bookmarks Develop Window Help
0:00
91%
6:21 PM
sspai.com
少数派
正版软件
付费栏目
Matrix
专题广场
热门文章
应用推荐
生活方式
新玩意
捷径：由浅入深完全指南
¥88
10 / 30
立即阅读
分享给好友
栏目信息

用 Keyboard Maestro 实现了双击退出 Safari

就「防止手贱退出应用」这个场景而言，还有其他的解决办法，如果你更喜欢长按而不是 Double Key，下面两篇文章可以提供参考。

- 《告别误触，防止「手抖」退出 macOS 应用：SlowQuitApps》
- 《CommandQ，让手速党不再因为误按而退出整个应用》

最后，Hyper 键也可以打造成 Double Key，这个就留给大家自己玩吧。

小结

快捷键虽然古老，但对于追求操作效率的人来说，它仍然是

一项基本功。

今天的键位已经和最初的样式大不相同。想要把快捷键背下来、用起来，就要摸出它们的设计思路；但仅仅学现成的键位，也是不够的，毕竟不可能一套快捷键就满足所有用户、适应所有工作，所以我们还要学会各种自定义的方法，把最趁手的操作方式，留给自己最需要的功能。

用快捷键，就像学一门功夫，知其然、亦知其所以然，更令其顺乎自然。